

**SCORING SYSTEM OTOMATIS PADA LOMBA MENEMBAK  
DENGAN TARGET *SILHOUETTE* HEWAN MENGGUNAKAN  
METODE *K-NEAREST NEIGHBOR (K-NN)***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh:  
Syahriel Diovanni Yolanda

NIM: 145150300111078



PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

SCORING SYSTEM OTOMATIS PADA LOMBA MENEMBAK DENGAN TARGET  
SILHOETTE MENGGUNAKAN METODE *K-Nearest Neighbor (K-NN)*

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun Oleh :  
Syahriel Diovanni Yolanda  
NIM: 145150300111078

Skripsi ini telah diuji dan dinyatakan lulus pada  
26 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Rizal Maulana, S.T., M.T., M.Sc.  
NIK: 201607 891009 1 001

Dosen Pembimbing II



Mochammad Hannats Hanafi Ichsan., S.ST., M.T  
NIK: 201405 881229 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan S.T., M.T., Ph.D  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Penulis menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Juli 2018



Syahriel Diovanni Yolanda

NIM: 145150300111078

## KATA PENGANTAR

Puja dan puji syukur penulis panjatkan kepada Allah SWT atas berkah, rahmat serta hidayah-Nya, penulis dapat menyelesaikan skripsi ini untuk memperoleh gelar Sarjana Teknik. Adapun judul skripsi yang disusun yakni "*Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Metode K-Nearest Neighbor (K-NN)*".

Dalam menyusun skripsi ini, banyak hambatan yang dihadapi oleh penulis. Penulis menyadari tanpa bantuan serta doa dari berbagai pihak, skripsi ini tidak dapat terselesaikan. Pada kesempatan kali ini, penghargaan dan terimakasih yang sebesar-besarnya penulis sampaikan kepada :

1. Ayah Osin dan Bunda Siti Islamiyah selaku orang tua penulis serta keluarga penulis yang telah memberikan doa dan dukungan.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. Selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Heru Nurwasito, Ir., M.Kom. Selaku Wakil Dekan Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. Selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
5. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. Selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
6. Bapak Rizal Maulana, S.T.,M.T.,M.Sc. Selaku Dosen Pembimbing I yang telah memberikan pengarahan dalam pembuatan skripsi ini.
7. Bapak Mochammad Hannats Hanafi Ichsan.,S.ST., M.T. Selaku Dosen Pembimbing II yang telah memberikan pengarahan dalam pembuatan laporan skripsi ini.
8. Seluruh dosen dan karyawan Fakultas Ilmu Komputer yang telah memberikan bantuan dan dukungan selama proses penyelesaian skripsi ini.

9. Sri Nila Agus Sari A.Md.Keb yang telah memberikan support , semangat serta doa kepada penulis.
10. Handoko Ramadhan., S.T. yang telah memberikan pengarahan dan motivasi kepada penulis.
11. Keluarga Besar Kontrakan Ceria diantaranya Nahja Falahil Umam , Wahyu Pria Purnama , Yongki Pratama , dan Hamim Fathul Aziz yang sudah memberi dukungan dan semangat kepada penulis.
12. Kerabat dan Sahabat Dekat D'Gengz diantaranya Adrian Mitra Perwira, Viramuda Tantri Burhan Mubarak , Intan Fatmawati, Ida yusnilawati , Putri Ayu Delina Sari , Linda Silvy Putri , Muzammilatul Jamiilah , Lita Nur Fitriani ,Yongki Pratama dan Aveiro Ilham yang selalu memberi motivasi kepada penulis.
13. Dulur-Dulur Teknik Komputer 2014 dan juga adik tingkat Teknik Komputer 2016 & 2017 yang selalu memberi dukungan kepada penulis.
14. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu yang telah berperan dalam penyelesaian skripsi ini.

Akhir kata, penulis menyadari bahwa skripsi ini tak lepas dari kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dan semoga skripsi ini dapat bermanfaat bagi semua pembaca.

Malang, 3 Juli 2018

Penulis

Syahrieldiovanni888@gmail.com



## ABSTRAK

Olahraga menembak adalah salah satu jenis olahraga. Olahraga menembak merupakan olahraga ketangkasan karena olahraga ini mampu meningkatkan tingkat fokus serta daya tangkap seseorang. Melalui olahraga menembak seseorang dilatih dan dituntut untuk dapat berkonsentrasi tinggi, mampu mengendalikan diri dan berani mengambil keputusan secara cepat dan tepat. Olahraga menembak merupakan latihan mengendalikan diri. Dalam menentukan keahlian seseorang dalam menembak, diperlukan sistem skoring yang akurat. Dalam sistem skoring penembakan pada umumnya dilakukan secara manual dibutuhkan pengamat/juri untuk menilai hasil tembakan. Sistem skoring secara manual membutuhkan waktu lebih dalam penyampaian informasinya sampai ke penonton. Pada penelitian ini akan dibuat *Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Silhouette* Hewan Menggunakan Metode *K-Nearest Neighbor* untuk menentukan jenis *Silhouette* mana yang tertembak dan menambahkan skor pada monitor sistem. Metode *K-Nearest Neighbor* digunakan dalam proses klasifikasi jenis *Silhouette* yang tertembak untuk memberi pengaruh terhadap skor dalam perlombaan. Pada sistem ini menggunakan beberapa komponen yaitu sensor Getar SW-420, *Load Cell*, *Limit Switch*, *Switch*, Modul *HX711*, serta *Arduino Mega 2560* sebanyak 2 sebagai pemrosesnya. Berdasarkan hasil uji coba pada sistem ini, didapatkan tingkat akurasi terhadap data sensor sebesar 96,42% dan tingkat akurasi terhadap klasifikasi *K-Nearest Neighbor* adalah sebesar 100%.

Kata kunci: *Scoring System* Otomatis , *Silhouette* Hewan , Metode Klasifikasi *K-Nearest Neighbor*

## **ABSTRACT**

Shooting Sport is a one type of sport. Sports shooting is a sport of agility because this sport can increase the level of focus and capture a person. Through sports shooting someone trained and required to be able to concentrate, able to control themselves and dare to make decisions quickly and accurately. Shooting is a self-control exercise. In determining one's expertise in firing, an accurate scoring system is required. In a generalized scoring system, manual shooting requires an observer / judge to assess the firing result. The scoring system manually takes more time in delivering the information to the audience. In this study we will create an Automated Scoring System On Shooting Contest With Animal Silhouette Target Using K-Nearest Neighbor Method to determine which type of Silhouette is shot and add score on the system monitor. The K-Nearest Neighbor method is used in the classification process of the shooting Silhouette type to give effect to the score in the race. In this system uses some components of vibration sensor SW-420, Load Cell, Limit Switch, Switch, Module HX711, and Arduino Mega 2560 as much as 2 processors. Based on test result on this system, the accuracy level of sensor data is 96,42% and accurate to K-Nearest Neighbor classification is 100%.

Keywords: Automatic Scoring System, Metal Silhouette, K-Nearest Neighbor Classification Method

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
<i>ABSTRACT</i> .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN .....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian .....	3
1.4 Manfaat Penelitian .....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Tinjauan Pustaka .....	6
2.2 Dasar Teori .....	8
2.2.1 Menembak .....	8
2.2.2 <i>Scoring System</i> .....	8
2.2.3 <i>Silhouette Hewan</i> .....	9
2.2.4 Sensor Getar.....	9
2.2.5 Sensor <i>Load Cell</i> .....	10
2.2.6 Modul HX711.....	10
2.2.7 <i>Limit Switch</i> .....	11
2.2.8 Arduino Mega 2560.....	11
2.2.9 Analog to Digital Conversion (ADC).....	12
2.2.10 <i>K-Nearest Neighbor (K-NN)</i> .....	13
BAB 3 METODOLOGI .....	16



3.1 Metode Penelitian .....	16
3.2 Studi Literatur .....	16
3.3 Analisa Kebutuhan .....	17
3.3.1 Kebutuhan Perangkat Keras.....	17
3.3.2 Kebutuhan Perangkat Lunak .....	18
3.4 Perancangan Sistem.....	18
3.5 Implementasi .....	19
3.5.1 Konfigurasi Perangkat Keras .....	19
3.5.2 Konfigurasi Perangkat Lunak.....	19
3.6 Pengujian dan Analisis .....	19
3.7 Kesimpulan.....	19
BAB 4 REKAYASA KEBUTUHAN.....	20
4.1 Gambaran Umum Sistem.....	20
4.2 Analisa Kebutuhan Sistem .....	20
4.2.1 Kebutuhan Fungsional.....	20
4.2.2 Kebutuhan Non-Fungsional .....	22
4.3 Batasan Desain Sistem .....	23
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	24
5.1 Perancangan Sistem.....	24
5.1.1 Perancangan Alat Mekanik, Peletakan dan Silhouette .....	24
5.1.2 Perancangan Perangkat Keras .....	25
5.1.3 Perancangan Perangkat Lunak.....	28
5.1.4 Perancangan Papan Skor.....	30
5.1.5 Perancangan <i>K-Nearest Neighbor (K-NN)</i> .....	30
5.2 Implementasi Sistem .....	32
5.2.1 Implementasi Alat Mekanik, Peletakan dan <i>Silhouette</i> .....	32
5.2.2 Implementasi Perangkat Keras .....	32
5.2.3 Implementasi Perangkat Lunak.....	33
5.2.4 Implementasi Papan Skor .....	33
5.2.5 Implementasi <i>K-Nearest Neighbor (K-NN)</i> pada Sistem <i>Scoring</i> .....	35
BAB 6 PENGUJIAN DAN ANALISIS.....	40
6.1 Pengujian Akuisisi Data Sensor Getar .....	40

6.1.1 Tujuan.....	40
6.1.2 Langkah-Langkah.....	40
6.1.3 Hasil dan Analisis.....	41
6.2 Pengujian Akuisisi Data <i>Load Cell</i> .....	47
6.2.1 Tujuan.....	47
6.2.2 Langkah-Langkah.....	47
6.2.3 Hasil dan Analisis.....	48
6.3 Pengujian Akuisisi Data <i>Limit Switch</i> .....	50
6.3.1 Tujuan Pengujian.....	50
6.3.2 Langkah-Langkah.....	51
6.3.3 Hasil dan Analisis.....	51
6.4 Pengujian Proses Klasifikasi <i>K-Nearest Neighbor</i> .....	52
6.4.1 Tujuan Pengujian.....	52
6.4.2 Langkah-Langkah.....	52
6.4.3 Hasil dan Analisis.....	52
BAB 7 PENUTUP .....	55
7.1 Kesimpulan.....	55
7.2 Saran .....	56
DAFTAR PUSTAKA.....	57
LAMPIRAN .....	58

## DAFTAR TABEL

Tabel 2.1 Perbandingan Paper .....	7
Tabel 5.1 Koneksi Pin Perancangan Perangkat Keras .....	27
Tabel 5.2 Tabel Code Program Klasifikasi <i>K-Nearest Neighbor</i> .....	35
Tabel 6.1 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Ayam Normal .....	41
Tabel 6.2 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Ayam Tersentuh.....	42
Tabel 6.3 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Ayam Tertembak.....	42
Tabel 6.4 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Kalkun Normal .....	43
Tabel 6.5 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Kalkun Tersentuh .....	43
Tabel 6.6 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Kalkun Tertembak.....	44
Tabel 6.7 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Babi Normal .....	44
Tabel 6.8 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Babi Tersentuh .....	45
Tabel 6.9 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Babi Tertembak.....	45
Tabel 6.10 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Domba Normal.....	46
Tabel 6.11 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Domba Tersentuh .....	46
Tabel 6.12 Hasil Pengujian Sensor Getar Jenis <i>Silhouette</i> Domba Tertembak.....	47
Tabel 6.13 Hasil Pengujian Sensor <i>Load Cell</i> Jenis <i>Silhouette</i> Ayam.....	48
Tabel 6.14 Hasil Pengujian Sensor <i>Load Cell</i> Jenis <i>Silhouette</i> Kalkun .....	49
Tabel 6.15 Hasil Pengujian Sensor <i>Load Cell</i> Jenis <i>Silhouette</i> Babi.....	49
Tabel 6.16 Hasil Pengujian Sensor <i>Load Cell</i> Jenis <i>Silhouette</i> Domba .....	50
Tabel 6.17 Hasil Pengujian <i>Limit Switch</i> .....	51
Tabel 6.18 Hasil Pengujian <i>K-Nearest Neighbor</i> .....	54

## DAFTAR GAMBAR

Gambar 2.1 <i>Silhouette</i> Hewan .....	9
Gambar 2.2 Sensor Getar .....	9
Gambar 2.3 <i>Load Cell</i> .....	10
Gambar 2.4 Modul HX711 .....	10
Gambar 2.5 <i>Limit Switch</i> .....	11
Gambar 2.6 Arduino Mega 2560 .....	12
Gambar 2.7 Resolusi Sampling ADC .....	12
Gambar 2.8 Diagram Alir <i>K-Nearest Neighbor</i> .....	15
Gambar 3.1 Diagram Alir Metode Penelitian .....	16
Gambar 3.2 Diagram Blok Rancangan Sistem .....	18
Gambar 5.1 Perancangan Alat Tampak Samping .....	24
Gambar 5.2 Perancangan Alat Tampak Atas .....	25
Gambar 5.3 Skema Perancangan Perangkat Keras .....	26
Gambar 5.4 Rangkaian Debounce .....	26
Gambar 5.5 Koneksi I2C Antar Arduino .....	27
Gambar 5.6 Flowchart Perancangan Perangkat Lunak .....	29
Gambar 5.7 Flowchart Perancangan Papan Skor .....	30
Gambar 5.8 Flowchart Perancangan <i>K-Nearest Neighbor</i> (K-NN) .....	31
Gambar 5.9 Implementasi Prototipe .....	32
Gambar 5.10 Implementasi Rangkaian Pada Box <i>Silhouette</i> .....	33
Gambar 5.11 Papan Skor pada Layar Monitor .....	34
Gambar 5.12 Menu <i>Setting</i> .....	34
Gambar 6.1 Pengujian dan Analisis .....	40
Gambar 6.2 Hasil Klasifikasi Dari Data Uji Coba .....	43

## DAFTAR LAMPIRAN

Lampiran 1 Data <i>Training</i> .....	58
---------------------------------------	----



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Olahraga merupakan suatu kebutuhan jasmani yang harus dipenuhi dan merupakan salah satu unsur yang berpengaruh dalam kehidupan manusia sejak dulu. Sejarah perkembangan olahraga pertama kali ditemukan pada zaman prasejarah sekitar 30.000 tahun yang lalu. Hal ini dibuktikan dengan adanya lukisan/gambar yang menampilkan beberapa aktivitas seperti renang dan memanah. Sedikit bukti yang didapat dari zaman prasejarah ini, namun cukup beralasan untuk menyimpulkan bahwa pada zaman prasejarah juga terdapat aktivitas yang berkenaan dengan olahraga. Perkembangan olahraga terus berkembang di Negara-negara besar lainnya, seperti Cina Kuno, Mesir Kuno, Yunani Kuno, dan Eropa. Seiring dengan perkembangan ilmu pengetahuan teknologi dan kebutuhan manusia, maka kegiatan olahraga tidak hanya dilakukan untuk mengekspresikan diri dan pemeliharaan diri melainkan untuk kegiatan olahraga yang menghibur dan dipertandingkan (Perlaita, 2010).

Salah satu jenis olahraga adalah olahraga menembak. Olahraga menembak merupakan olahraga ketangkasan karena olahraga ini mampu meningkatkan tingkat fokus serta daya tangkap seseorang. Menembak juga membutuhkan suatu konsentrasi yang tinggi. Melalui olahraga menembak seseorang dilatih dan dituntut untuk dapat berkonsentrasi tinggi, mampu mengendalikan diri dan berani mengambil keputusan secara cepat dan tepat (Bambang, 2008).

Olahraga menembak yakni menembak target, menembak reaksi dan menembak berburu. Seorang petembak yang baik adalah yang dapat sepenuhnya mengontrol dirinya sendiri baik mental maupun fisiknya. Olahraga menembak adalah latihan mengendalikan diri. Disiplin dalam berlatih dan mengulang keberhasilan adalah kunci keberhasilan seorang petembak. Dalam menentukan keahlian seseorang dalam menembak, diperlukan sistem skoring yang akurat. Dalam sistem skoring penembakan pada umumnya dilakukan secara manual dibutuhkan pengamat/juri untuk menilai hasil tembakan. Sistem skoring secara manual membutuhkan waktu lebih dalam penyampaian informasinya sampai ke penonton (Farida dan Hermana, 2007).

Perkembangan teknologi informasi (TI) telah mendorong peningkatan Efisiensi pelatihan dalam olahraga. Jika sistem pelatihan dikembangkan dengan cara terintegrasi antara TI dengan teknologi otomatis lainnya. Keterampilan atlet dalam berolahraga akan lebih meningkat. Jika dalam suatu pelatihan olahraga, informasi hasil latihan memberi umpan balik yang tepat (S. Ying, W. Gang, & W. Yaojun, 2011).

Oleh karena itu perlu adanya sistem yang mampu mengelola hasil skoring secara otomatis pada kegiatan berolahraga khususnya menembak. Sistem skoring yang dibuat disesuaikan dengan kebutuhan yang ada pada olahraga menembak. Dengan menggunakan perangkat Papan Skor Otomatis (PSO) dalam berolahraga menembak, maka akan membantu untuk mengevaluasi dan meningkatkan hasil



kinerja dari atlet menembak, dan memodernisasi *venue* menembak yang ada (Harmaen, 2012).

Penelitian kali ini merupakan sebuah pengembangan dari prototipe sistem skoring menembak otomatis yang telah dibuat sebelumnya. Berdasarkan hasil uji coba menembak dengan *sillhouette* berbentuk hewan, dibandingkan dengan cara manual yang dilakukan oleh petugas, sistem skoring yang dibuat memberikan hasil yang baik. Tingkat akurasi sistem yang dibuat sebesar 94.97% pada akurasi data sensornya. Sistem juga dapat mengklasifikasi jenis *sillhouette* menggunakan logika fuzzy dengan tingkat akurasi sebesar 100% (Handoko, 2017).

Sistem skoring otomatis ini menggunakan target yang akan digunakan berupa metal *silhouette*. Metal *silhouette* merupakan kegiatan menembak sebuah benda mati atau target berupa gambar atau benda yang mirip dengan target hewan yang diburu dalam bentuk mini, kegiatan ini biasanya dilaksanakan pada lomba-lomba menembak. Pada umumnya yang di pergunakan pada latihan olahraga menembak yakni kelas senapan angin yang berupa jenis pompa, per dan gas (pcp), dan jarak yang dilombakan sejauh 10 meter tanpa menggunakan *scope* sedangkan yang menggunakan *scope* bisa sejauh 25-50 meter tergantung dari kelas senjata yang di gunakan. Sistem skoring otomatis yang akan di buat berawal dari inisialisasi target dari masing-masing *silhouette*. Setelah masing-masing *silhouette* sudah diinisialisasi maka target siap untuk di tembak. Saat terjadi tembakan pada target *silhouette* maka sistem akan membaca nilai dari sensor getar dan berat untuk mengetahui target apakah yang tertembak dan hasil dari tembakan tersebut. Jika target yang di tembak terjatuh dan sesuai dengan target yang dituju maka system akan memberikan nilai sebesar 10 poin untuk target yang tertembak.

Kesulitan dalam suatu penilaian/skor penembakan dalam latihan olahraga menembak merupakan salah satu alasan dibuat penelitian yang menyangkut sistem skoring otomatis dengan menggunakan metode *K-Nearest Neighbor (K-NN)*. Penelitian ini menggunakan metode *K-Nearest Neighbor (K-NN)* dikarenakan metode ini merupakan algoritma *supervised* yang dimana hasil dari sample uji yang baru di klasifikasikan berdasarkan mayoritas dan kategori pada K-NN dengan mengklasifikasikan objek baru berdasarkan atribut dan data *training* (Kusrini dan Luthfi, 2009).

Dengan menggunakan metode ini dapat membuat prototipe yang sesuai dengan apa yang diujikan. Dikarenakan metode ini sangat relevan dengan data *training* dibandingkan menggunakan metode pengklasifikasian yang lainnya. Jika semakin banyak data *training* yang dihasilkan maka *output* yang mendekati dari data *training* tersebut kemungkinan besar hasil targetnya adalah sama. Jadi penggunaan metode ini memiliki keakuratan yang maksimal dibandingkan metode yang lain dalam pembuatan prototipe ini.

Ketepatan algoritma *K-NN* ini juga sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik. K buah data learning terdekat akan melakukan voting untuk menentukan label mayoritas. (Bronshtein, 2017)

Hal ini yang membuktikan bahwa metode ini lebih baik dari pada metode lainnya karena metode ini memilih secara acak di antara label-label mayoritas yang ada. Label data query akan ditentukan berdasarkan label mayoritas dan jika ada lebih dari satu label mayoritas maka label data query dapat dipilih secara acak di antara label-label mayoritas yang ada. Sehingga hasil data query tetap di pilih dari label mayoritas yang ada tanpa harus melakukan perbandingan terhadap semua data query yang ada, namun hal ini terjadi ketika label mayoritas harus berjumlah lebih dari satu. Jika hanya terdapat satu label mayoritas maka hasil yang di pilih terdapat pada label mayoritas itu sendiri.

Penelitian ini menggunakan sebuah perangkat mikrokontroler yaitu Arduino Mega 2560. Mikrokontroler Arduino Mega 2560 digunakan karena mikrokontroler ini mempunyai memori yang lebih besar di bandingkan mikrokontroler Arduino versi lainnya dan juga dalam pengimplementasian metode *K-Nearest Neighbor* (K-NN) dalam mikrokontroler membutuhkan data *training* yang banyak dan nantinya dijadikan acuan dalam pengklasifikasian data yang akan di ujikan, maka dari itu membutuhkan banyak memori untuk memprosesnya serta memiliki banyak pin yang dapat memudahkan *user* untuk menggunakan banyak komponen sekaligus. Pada sistem ini, peneliti menggunakan sensor getar (SW-420) yang berfungsi untuk menangkap getaran pada saat tembakan mengenai target dan sensor berat (*Load Cell*) yang berfungsi menangkap berat dari tiap target metal *silhouette*. Kemudian data yang di dapat dari sensor – sensor tersebut akan dikirim dengan pengiriman data antar kontroler menggunakan Kabel.

## 1.2 Rumusan Masalah

Dengan permasalahan yang dihadapi, penulis melakukan perumusan masalah sebagai berikut :

1. Bagaimana merancang sistem *scoring* otomatis pada lomba menembak dengan target *silhouette* hewan ?
2. Bagaimana mengimplemetasikan Metode *K-Nearest Neighbor* (K-NN) terhadap sistem *scoring* otomatis pada lomba menembak dengan target *silhouette* hewan ?
3. Bagaimana hasil yang didapatkan pada Metode *K-Nearest Neighbor* (K-NN) dari sistem *scoring* otomatis pada lomba menembak dengan target *silhouette* hewan ?

## 1.3 Tujuan Penelitian

Dengan dirumuskannya masalah, didapatkan tujuan yang terkait pada rumusan masalah sebagai berikut :

1. Melakukan perancangan dan pengimplementasikan sistem *scoring* otomatis pada lomba menembak dengan target metal *silhouette* hewan
2. Melakukan penerapan Metode *K-Nearest Neighbor* (K-NN) terhadap sistem *scoring* otomatis pada lomba menembak dengan target metal *silhouette* hewan

3. Mengetahui kesesuaian Metode *K-Nearest Neighbor (K-NN)* terhadap sistem *scoring* otomatis pada lomba menembak dengan target metal *silhouette* hewan

#### 1.4 Manfaat Penelitian

Dilihat dari fungsi sistem yang akan diteliti, didapatkan manfaat sebagai berikut :

1. Dapat membantu atlet cabang olahraga menembak, dengan sistem scoring yg otomatis pada saat melakukan sesi latihan.
2. Keuntungan bagi penyelenggara lomba menembak dengan sistem scoring otomatis.

#### 1.5 Batasan Masalah

Adapun agar pembahasan menjadi lebih terarah dan tidak menyimpang sesuai dengan latar belakang yang telah diuraikan. Oleh karena itu penulis melakukan pembatasan penelitian yang dapat dicantumkan sebagai berikut :

1. Target yang digunakan berupa metal *silhouette* yang berbentuk seperti hewan sejumlah 4 buah *silhouette*.
2. Delphi yang akan digunakan sebagai tampilan informasi nilai skor pada perlombaan menembak.
3. Target penembakan yang digunakan adalah penembakan target secara *random* atau acak.

#### 1.6 Sistematika pembahasan

Guna memahami lebih jelas pembahasan proposal skripsi ini, dilakukan dengan cara mengelompokkan materi menjadi beberapa sub bab dengan sistematika pembahasan sebagai berikut :

##### **BAB I : PENDAHULUAN**

Bab ini menjelaskan tentang informasi umum yaitu latar belakang penelitian, perumusan masalah, tujuan dan manfaat penelitian, batasan penelitian, dan sistematika dari penelitian yang akan dilakukan yaitu "*Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Metode K-Nearest Neighbor (K-NN)*"

##### **BAB II : LANDASAN KEPUSTAKAAN**

Bab ini menguraikan landasan kepastakaan dari tugas akhir ini yang akan dibuat yaitu "*Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Metode K-Nearest Neighbor (K-NN)*". Bab ini juga menjelaskan konsep dasar *K-Nearest Neighbor (K-NN)*, sensor getar, *load cell*, *limit switch* dan NRF24L01 pada sistem alat, dan informasi lain yang berkaitan dengan batasan masalah yang peneliti berikan.

**BAB III : METODE PENELITIAN**

Bab ini menjelaskan alur kerja yang akan digunakan dalam penelitian ini. Alur kerja yang digunakan pada penelitian adalah studi literatur, analisis kebutuhan sistem yang terdiri dari kebutuhan fungsional, kebutuhan perangkat keras dan kebutuhan perangkat lunak, perancangan sistem yang dijelaskan dengan blok diagram, dan implementasi dari metode Metode *K-Nearest Neighbor (K-NN)* kedalam sistem untuk pengambilan keputusan *output*.

**BAB IV : REKAYASA KEBUTUHAN**

Bab ini membahas mengenai persyaratan yang dibutuhkan dalam membangun *prototype scoring system* otomatis pada lomba menembak dengan target *Silhouette* hewan menggunakan metode *K-Nearest Neighbor (K-NN)*, kebutuhan sistem, software dan hardware.

**BAB V : PERANCANGAN DAN IMPLEMENTASI**

Bab ini membahas mengenai perancangan *prototype scoring system* otomatis pada lomba menembak dengan target *Silhouette* hewan menggunakan metode *K-Nearest Neighbor (K-NN)*.

**BAB VI : PENGUJIAN DAN ANALISA**

Bab ini membahas mengenai pengujian dan analisa terhadap *prototype scoring system* otomatis pada lomba menembak dengan target *Silhouette* hewan menggunakan metode *K-Nearest Neighbor (K-NN)*.

**BAB VII : PENUTUP**

Bab ini memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat keras dan lunak yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.

## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Tinjauan Pustaka

Pada penelitian sebelumnya yang telah dibuat oleh Handoko Ramadhan dengan judul *Scoring System Otomatis pada Lomba Menembak dengan Target Sillhouette Hewan Menggunakan Logika Fuzzy*. Logika fuzzy digunakan pada proses klasifikasi jenis Sillhouette untuk mempengaruhi skor pada perlombaan. Prototype ini dibangun menggunakan beberapa modul yaitu Load Cell, Sensor Getar, Limit Switch, Modul HX711, dan menggunakan Arduino Uno serta Arduino Mega 2560 sebagai pemrosesnya. Berdasarkan hasil uji coba menembak dengan Sillhouette berbentuk hewan, dibandingkan dengan cara manual yang dilakukan oleh petugas, sistem skoring ini memberikan hasil yang baik. Tingkat akurasi sistem ini sebesar 94.97% pada akurasi data sensornya. Sistem juga dapat mengklasifikasi jenis Sillhouette menggunakan logika fuzzy dengan tingkat akurasi sebesar 100% (Ramadhan, Maulana, & Ichsan, 2018).

Pada tahun 2017 Nanda Epriliana telah meneliti terkait dengan sistem klasifikasi untuk mendeteksi status gizi pada bayi, dengan judul *Implementasi K-Nearest Neighbor pada Embedded System untuk Menentukan Status Gizi Bayi*. Pada sistem ini membutuhkan parameter-parameter berupa jenis kelamin, umur, panjang dan berat badan bayi. Dalam pembuatan sistem menggunakan komponen elektronika berupa modul, sensor dan mikrokontroler. Berdasarkan permasalahan tersebut, dibuatlah sistem klasifikasi status gizi bayi dengan metode K-Nearest Neighbor berbasis sistem embedded berdasarkan parameter jenis kelamin, umur, dan berat badan dari bayi. Pemilihan nilai parameter K yang akan diterapkan pada metode perlu adanya pencarian nilai dengan uji coba terlebih dahulu agar mendapatkan hasil nilai K terbaik untuk diterapkan pada sistem. Untuk pembacaan data berat menggunakan rangkaian dengan satu modul HX711, satu sensor load cell, yang dihubungkan dengan NodeMCU ESP8266 agar hasil pembacaan sensor dapat dikirim secara wireless ke komputer/PC yang akan digunakan sebagai parameter pengklasifikasian. Masukan data berupa jenis kelamin dan umur juga dibutuhkan untuk pengklasifikasian, sehingga jika ketiga parameter sudah terpenuhi hasil status gizi dapat langsung ditampilkan dengan menekan tombol Lihat Hasil. Selanjutnya semua data yang ditampilkan pada web dapat disimpan ke dalam basis data sehingga bisa digunakan untuk arsip data. Dari analisis yang dilakukan, didapatkan kesimpulan bahwa pengujian fungsional pada sensor berat memiliki persentase akurasi sebesar 97,23%, sedangkan pengujian fungsional penyimpanan basis data memiliki persentase kesesuaian 100%. Untuk pengujian pencarian nilai K didapatkan persentase akurasi tertinggi ketika K=5 dan k=6 dengan besar persentase 62,50%. Sedangkan pengujian keseluruhan sistem klasifikasi status gizi bayi menghasilkan akurasi sebesar 97,14%. (Epriliana, Dahnia, & Ichsan, 2017)



Pelana dan Hanif telah membuat penelitian yang bertujuan untuk mengembangkan alat penampil skor otomatis pada papan target dalam olahraga panahan. Pada penelitian tersebut melakukan pemanfaatan mikrokontroler AT89S51 untuk membentuk suatu minimum sistem sebagai alat penampil skor otomatis. Komunikasi antar sistem yang menggunakan mikrokontroler AT89S51 dengan sensor pada papan target dan display skor adalah suatu komunikasi berupa pengiriman data dimana output sensor pada papan target sebagai input, mikrokontroler sebagai pengolah data. Komunikasi yang dilakukan pada papan displa skor dan total skor berupa pengiriman data biner BCD dari mikrokontroler AT89S51, yang kemudian dikodekan dalam Sevent Segment (Pelanda & Hanif ,2013).

Berdasarkan pada tiga referensi penelitian yang telah dilakukan diatas, terdapat kemiripan pada objek penelitian yaitu berupa jenis sensor yang digunakan, hingga cara kerja sistem yang digunakan. Sehingga peneliti memilih tiga referensi tersebut sebagai referensi penelitian. Penelitian yang dilakukan memiliki objek penelitian menggunakan perancangan *scoring system* otomatis pada lomba menembak dengan target *Silhouette* hewan. Sensor yang digunakan pada penelitian penulis menggunakan sensor getar, *limit switch*, *load cell*.

**Tabel 2.1** Perbandingan Paper

No.	Nama Peneliti	Perbedaan	
		Penelitian Terdahulu	Rencana Penelitian
1	(Ramadhan, Maulana, & Ichsan, 2018).	Scoring System Otomatis pada Lomba Menembak dengan Target Silhouette Hewan Menggunakan Logika Fuzzy dengan menggunakan Arduino Mega 2560 dan sensor getar dan sensor load cell	Menggunakan Mikrokontroler Arduino Mega 2560 dan sensor getar SW-420 dan sensor berat ( <i>Load Cell</i> )
2	(Epriliana, Dahnia, & Ichsan, 2017)	Implementasi <i>K-Nearest Neighbor</i> pada Embedded System untuk Menentukan Status Gizi Bayi menggunakan rangkaian dengan satu modul <i>HX711</i> , satu sensor <i>load cell</i> , yang dihubungkan dengan NodeMCU ESP8266.	Menggunakan Arduino Mega 2560 sebagai pemroses dan <i>K-Nearest Neighbor</i> sebagai metode klasifikasi



No.	Nama Peneliti	Perbedaan	
		Penelitian Terdahulu	Rencana Penelitian
3	(Pelanda & Hanif, 2013)	Skor otomatis pada papan target dalam olahraga panahan menggunakan mikrokontroler AT89S51 dengan sensor pada papan target dan display skor dengan pengiriman data biner BCD.	Skor otomatis pada papan target dalam olahraga menembak menggunakan Delphi sebagai output Skor yang akan di tampilkan dan kondisi jenis target yang ditembak

## 2.2 Dasar Teori

### 2.2.1 Menembak

Menembak adalah olahraga yang dapat membentuk kepribadian seseorang. Melalui olahraga menembak seseorang dilatih dan dituntut untuk dapat berkonsentrasi tinggi, mampu mengendalikan diri dan berani mengambil keputusan secara cepat dan tepat. Olahraga menembak baik itu menembak target, menembak reaksi dan menembak berburu. Secara harfiah, kata menembak berarti dua hal yang pertama yaitu, melepaskan peluru dari senapan api dan kedua, menunjukan, mengarahkan sesuatu kepada sesuatu. Dari kedua kata itu maka akan muncul tiga hal penting dari konsep menembak yang pertama ialah, kebendaan yaitu alat dari menembak itu. Kedua, yakni manusia yang merupakan subyek dari pemakaian alat itu dan ketiga target sebagai aktifitas obyek dari menembak melalui alat senapan atau pistol. Dari ketiga pengertian konsep menembak itulah maka dapat dilihat bahwa menembak merupakan suatu kerja ide dan indera yang berhimpun dalam suatu waktu, suatu tempat dan suatu reaksi, semua hal tersebut terakumulasi dalam kerja menembak tersebut. Dibanding cabang olahraga lain, olahraga menembak utamanya menembak reaksi merupakan cabang olahraga yang memerlukan kesinambungan kerja yang baik antara aksi dan reaksi. Dalam menembak, setiap penembak harus memiliki ketenangan, ketahanan dan pengontrolan diri (Herlambang, 2003).

### 2.2.2 Scoring System

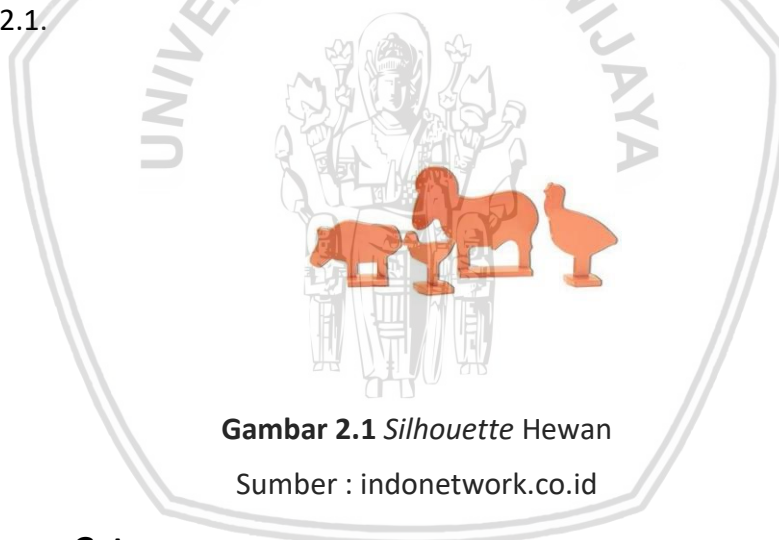
Diagnosis suatu proses pengukuran atribut adalah pemberian makna atau interpretasi terhadap skor skala yang bersangkutan. Sebagai suatu hasil ukur berupa angka (kuantitatif), *scoring system* yang disebut sebagai skor skala, memerlukan suatu norma pembandingan agar dapat diinterpretasikan secara kualitatif. Pada dasarnya, interpretasi skor skala selalu bersifat normative, artinya makna skor diacukan pada posisi relatif skor dalam suatu kelompok yang telah dibatasi terlebih dahulu. Deskripsi data ini memberi gambaran penting mengenai keadaan distribusi skor skala pada kelompok subjek yang dikenai pengukuran dan

berfungsi sebagai sumber informasi mengenai keadaan subjek pada aspek variabel yang diteliti (Wardhani, 2005).

Sedangkan untuk menerapkan metode *Scoring System* ke dalam sistem. Suatu skor yang ditentukan melalui prosedur penskalaan akan menghasilkan angka-angka pada level pengukuran interval dan interpretasikan hanya dapat dihasilkan kategori-kategori atau kelompok-kelompok skor pada level ordinal. Skor-skor mentah (row score) yang dihasilkan suatu skala merupakan penjumlahan dari skor item-item dalam skala itu. Skala penambahan skor yang digunakan setiap target yang tertembak dan sesuai target yang dituju yakni sebesar 10.

### 2.2.3 *Silhouette* Hewan

*Silhouette* hewan adalah sebuah benda mati atau target berupa gambar atau benda yang mirip dengan target hewan yang akan diburu dalam bentuk mini. Pada lomba menembak, jenis hewan *silhouette* yang digunakan, mempengaruhi jarak sasaran, jarak antar sasaran, dan tinggi sasaran. Bentuk *Silhouette* terdiri dari *silhouette* ayam, kalkun, babi dan domba. Untuk tiap *silhouette* mempunyai berat masing-masing yakni ayam sebesar 29 gram, kalkun sebesar 36 gram, Babi sebesar 48 gram, Domba sebesar 68 gram. Bentuk *silhouette* bisa dilihat pada Gambar 2.1.



**Gambar 2.1** *Silhouette* Hewan

Sumber : indonetwork.co.id

### 2.2.4 Sensor Getar

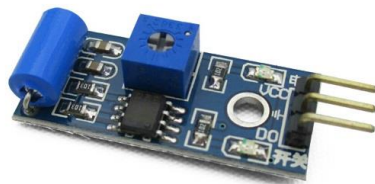
Sensor getar adalah suatu alat yang berfungsi untuk mendeteksi adanya getaran dan akan diubah dalam ke dalam sinyal listrik. Sensor ini disebut juga casing measurement. Sensor yang digunakan adalah sensor *seismic transducer*, yaitu sensor yang digunakan untuk mengukur kecepatan dan percepatan. (Putra, 2016)

Sensor ini dapat dihubungkan dengan mikrkontroler dan memiliki tiga buah pin, yaitu vcc(+), DO (Output data), dan ground (-). Data output sensor ini (diakses dalam : depoinovasi, 2017). Spesifikasi sensor ini adalah :

1. Vsuplai : DC 3.3V-5V
2. Arus : 15mA
3. Sensor : SW-420 Normally Closed

4. Output : digital
5. Dimensi : 3,8 cm x 1,3 cm x 0,7 cm

Bentuk Sensor Getar bisa dilihat pada Gambar 2.2.



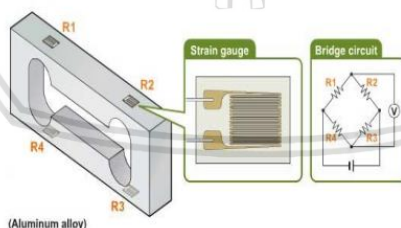
**Gambar 2.2** Sensor Getar

Sumber : [www.boarduino.web.id](http://www.boarduino.web.id)

### 2.2.5 Sensor Load Cell

*Load Cell* merupakan sensor berat, apabila *load cell* diberi beban pada inti besinya, maka nilai resistansi di *strain gauge* akan berubah. *Load cell* adalah alat elektromekanik yang biasa disebut Transducer, yaitu gaya yang bekerja berdasarkan prinsip deformasi sebuah material akibat adanya tegangan mekanis yang bekerja, kemudian merubah gaya mekanik menjadi sinyal listrik.

Sebuah *load cell* terdiri dari konduktor, *strain gauge*, dan jembatan *wheatstone*. *Strain gauge* merupakan *grid metal foil* tipis yang dilekatkan pada permukaan dari *load cell*. Apabila *load cell* di beri beban, maka terjadi strain induksi beban. Namun tegangan keluaran dari *load cell* sangat kecil, sehingga untuk mengetahui perubahan tegangan keluaran secara linier dibutuhkan rangkaian penguat *instrument* yang dapat menguatkan tegangan keluaran yang sangat kecil hingga kurang dari satuan *microvolt* (Rahmatullah, 2016). Bentuk Sensor Getar bisa dilihat pada Gambar 2.3.



**Gambar 2.3** Load Cell

Sumber : [www.kitomaindonesia.com](http://www.kitomaindonesia.com)

### 2.2.6 Modul HX711

Modul HX711 merupakan modul timbangan yang mempunyai prinsip kerja dapat mengkonversi perubahan yang terukur dengan adanya perubahan resistansi dan dapat mengkonversinya ke dalam besaran tegangan melalui rangkaian yang ada. Modul ini memiliki struktur yang sederhana, dan mudah dalam penggunaan,

dan mempunyai hasil yang stabil dan *reliable*. Spesifikasi modul ini adalah sebagai berikut :

- *Differential input voltage* :  $\pm 40\text{mV}$  (*Full-scale differential input voltage*  $\pm 40\text{mV}$ )
- *Data accuracy* : 24 bit (24bit A/D converter chip.)
- *Refresh Frequency* : 80 Hz
- *Operating Voltage* : 5V DC
- *Operating current* :  $<10\text{ mA}$
- *Size* : 38mm\*21mm\*10mm

Bentuk Modul HX711 bisa dilihat pada Gambar 2.4.



**Gambar 2.4** Modul HX711

Sumber : sparkfun.com

### 2.2.7 Limit Switch

*Limit switch* adalah saklar tekan yang berfungsi sebagai pemutus atau penyambung arus listrik dari sumber arus ke beban listrik. Suatu sistem saklar tekan *limit switch* terdiri dari saklar tekan start, stop reset dan saklar tekan untuk emergency. *Limit switch* memiliki kontak NC (normally close) dan NO (normally open). Bentuk *Limit Switch* bisa dilihat pada Gambar 2.5.



**Gambar 2.5** Limit Switch

Sumber : sparkfun.com

### 2.2.8 Arduino Mega 2560

Arduino Mega 2560 adalah sebuah mikrokontroler yang berbasis pada ATmega2560 dan merupakan inovasi baru dari board Arduino terdahulunya yaitu Arduino Mega 2560 dengan chip ATmega1280. Arduino Mega 2560 dapat dihubungkan dengan komputer menggunakan kabel USB dan dapat diprogram

dengan mudah karena Arduino kompatible dengan bahasa pemrograman C atau C++ yang di Custom menjadi sebuah *software* khusus sebagai platform pemrograman untuk Arduino.

Arduino Mega memiliki keunggulan dari segi source penyimpanan yang lebih besar daripada Arduino Uno dan Nano, selain itu jumlah digital pin yang cukup banyak (54 pin digital) dapat memudahkan *user* untuk menggunakan banyak komponen sekaligus. Berikut adalah spesifikasi Arduino Mega:

1. *Operating Voltage* :5V
2. *Input Voltage (recommended)* : 7-12V
3. *Input Voltage (limits)* : 6-20V
4. *Digital I/O Pins* : 54 (of which 15 provide PWM output)
5. *Analog Input Pins* : 16
6. *DC Current per I/O Pin* : 40 mA
7. *DC Current for 3.3V Pin* : 50 mA
8. *Flash Memory* : 256 KB of which 8 KB used by bootloader
9. SRAM : 8 KB
10. EEPROM : 4 KB
11. *Clock Speed* : 16 MHz

Bentuk Arduino Mega 2560 bisa dilihat pada Gambar 2.6.



**Gambar 2.6** Arduino Mega 2560

Sumber : [www.boarduino.web.id](http://www.boarduino.web.id)

### 2.2.9 Analog to Digital Conversion (ADC)

Analog To Digital Converter (ADC) adalah pengubah input analog menjadi kode digital. ADC sering digunakan untuk pengolahan data sensor analog dimana data outputnya biasanya berupa tegangan atau resistansi dimana rata-rata output tersebut masih belum bilangan bulat. Di dalam ADC data analog tersebut diubah menjadi data sampling dengan resolusi digital yang berbeda-beda. Semakin besar resolusi sampling data output maka semakin presisi pula output yang akan dihasilkan (Suprianto, 2015). Rumus dari ADC dapat dilihat pada **Persamaan (2.1)** berikut.

Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat



dinyatakan dalam 255 ( $2^n - 1$ ) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital, ini berarti sinyal input dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 bit.

$$ADC = \frac{sampel}{\max\_sampel} \times Range\ ADC \quad (2.1)$$

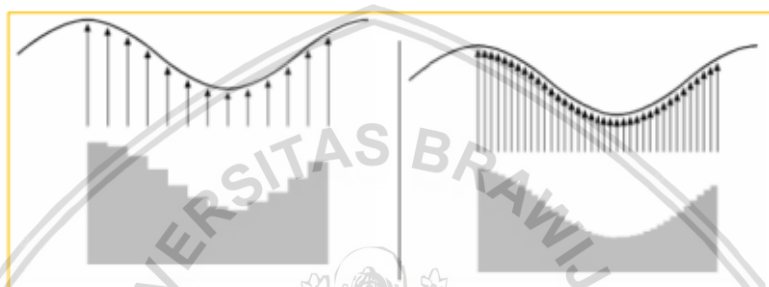
ADC : Nilai Analog to Digital Conversion (ADC)

Sampel : Nilai output sampel data analog

Max\_sampel : Nilai maksimum sampel data analog

Range ADC: Nilai ADC maksimum

Bentuk Resolusi Sampling ADC bisa dilihat pada Gambar 2.7.



**Gambar 2.7** Resolusi Sampling ADC

Sumber : elektronika-dasar.web.id

Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Sebagai contoh: ADC 8 bit akan memiliki output 8 bit data digital, ini berarti sinyal input dapat dinyatakan dalam 255 ( $2^n - 1$ ) nilai diskrit. ADC 12 bit memiliki 12 bit output data digital.

### 2.2.10 K-Nearest Neighbor (K-NN)

Algoritma *K-Nearest Neighbor* (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. *K-Nearest Neighbor* berdasarkan konsep ‘learning by analogy’. Data learning dideskripsikan dengan atribut numerik n-dimensi. Tiap data learning merepresentasikan sebuah titik, yang ditandai dengan c, dalam ruang n-dimensi. Jika sebuah data query yang labelnya tidak diketahui diinputkan, maka *K-Nearest Neighbor* akan mencari k buah data learning yang jaraknya paling dekat dengan data query dalam ruang n-dimensi. Jarak antara data query dengan data learning dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data query dengan semua titik yang merepresentasikan data learning dengan rumus Euclidean Distance. Pada fase training, algoritma ini hanya melakukan penyimpanan dan klasifikasi data training sample. Pada fase klasifikasi, fitur – fitur yang sama dihitung untuk testing data (klasifikasinya belum diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor training sample dihitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik – titik



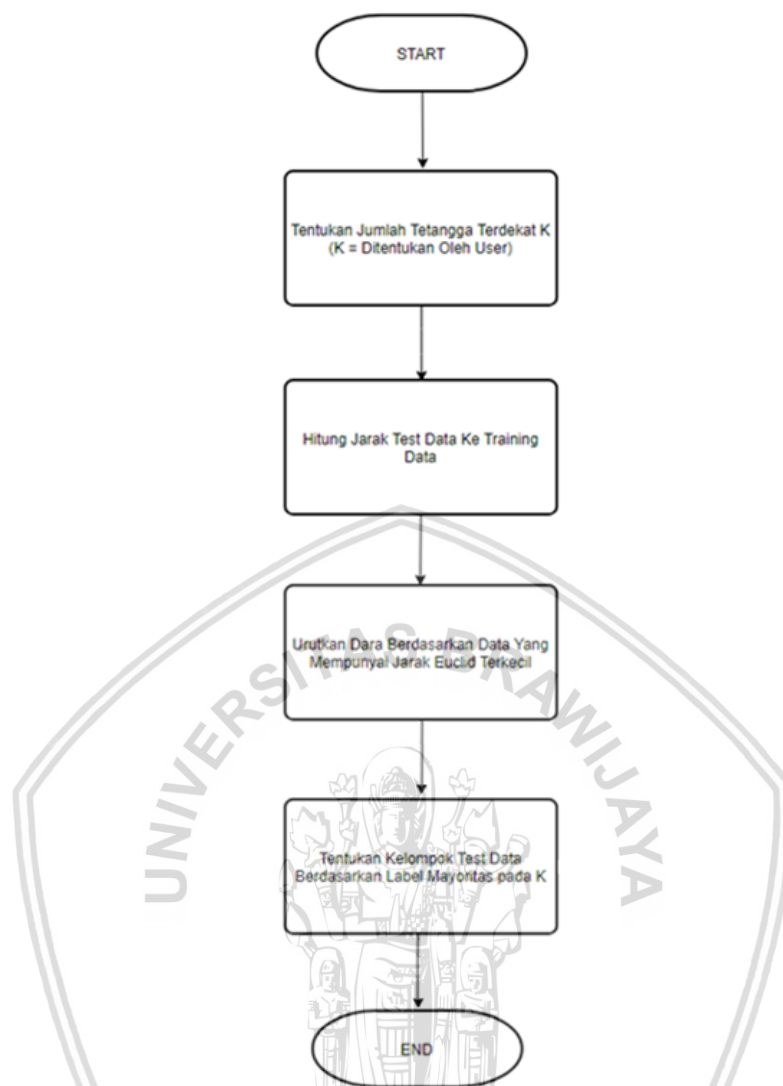
tersebut. Nilai  $k$  yang terbaik untuk algoritma ini tergantung pada data secara umumnya, nilai  $k$  yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai  $k$  yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain,  $k = 1$ ) disebut algoritma nearest neighbor (Kusrini dan Luthfi, 2009).

Ketepatan algoritma *K-NN* ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik.  $K$  buah data learning terdekat akan melakukan voting untuk menentukan label mayoritas. Label data query akan ditentukan berdasarkan label mayoritas dan jika ada lebih dari satu label mayoritas maka label data query dapat dipilih secara acak di antara label-label mayoritas yang ada.

Perhitungan Algoritma *K-Nearest Neighbor (K-NN)* merupakan sebuah metode untuk melakukan klasifikasi terhadap obyek baru berdasarkan ( $K$ ) tetangga terdekatnya (Gorunescu, 2011). *K-NN* termasuk algoritma supervised learning, dimana hasil dari query instance yang baru, diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Kelas yang paling banyak muncul yang akan menjadi kelas hasil klasifikasi. Langkah-langkah algoritma *K-NN* :

1. Menentukan parameter  $K$  (jumlah tetangga paling dekat), Parameter  $K$  pada testing ditentukan berdasarkan nilai  $K$  optimum pada saat training. Nilai  $K$  optimum diperoleh dengan mencoba-coba
2. Menghitung kuadrat jarak euclid (euclidean distance) masing-masing obyek terhadap data sampel yang diberikan
3. Mengurutkan objek-objek tersebut ke dalam kelompok yang mempunyai jarak euclidian terkecil
4. Dengan menggunakan kategori mayoritas, maka dapat hasil klasifikasi

Sebelum lanjut ke perhitungan algoritma *K-NN* perlu diketahui karakteristik dataset dari algoritma ini. Jadi dataset yang bisa di mining menggunakan algoritma ini adalah dataset yang mempunyai class/target/label nominal (tidak bisa dijumlahkan). Adapun algoritma dari *K-NN* ditunjukkan pada gambar 2.8 pada flowchart dibawah ini :



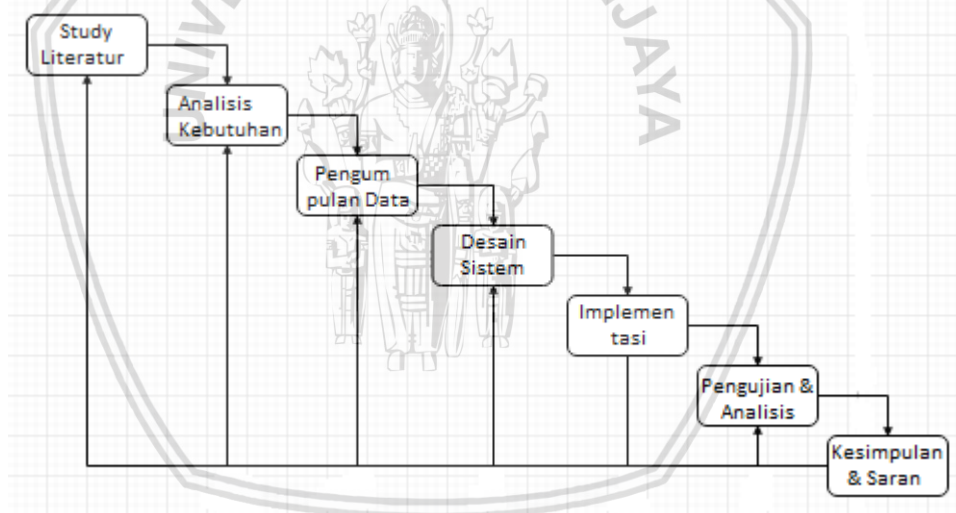
**Gambar 2.8** Diagram Alir *K-Nearest Neighbor*

Pada diagram alir *K-Nearest Neighbor* diatas menunjukan bahwa proses pengklasifikasian *K-Nearest Neighbor* memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik. K buah data learning terdekat akan melakukan voting untuk menentukan label mayoritas. Label data query akan ditentukan berdasarkan label mayoritas dan jika ada lebih dari satu label mayoritas maka label data query dapat dipilih secara acak di antara label-label mayoritas yang ada.

## BAB 3 METODOLOGI

### 3.1 Metode Penelitian

Bab ini menjelaskan metode yang digunakan dalam melakukan penelitian tentang perancangan *scoring system* otomatis pada lomba menembak dengan target metal *silhouette* hewan menggunakan metode *K-Nearest Neighbor (K-NN)*. Tipe penelitian yang dilakukan adalah implementatif. Implementatif yaitu bersifat observasi dengan menggunakan rancang bangun yang sederhana. Langkah pertama yang dilakukan adalah mengumpulkan teori-teori pendukung penelitian dan dikemas dalam studi literatur. Kemudian dilanjutkan dengan proses analisis kebutuhan. Setelah tahap analisis kebutuhan, proses selanjutnya adalah melakukan proses perancangan yang dilanjutkan dengan proses implementasi *hardware* serta implementasi *software* sesuai dengan perancangan. Tahap berikutnya dilakukan pengujian dan analisis pada rancangan yang telah dibangun. Kesimpulan dan saran disertakan sebagai catatan atas rancangan dan kemungkinan arah pengembangan selanjutnya. Langkah-langkah yang dilakukan dalam penelitian ini ditunjukkan Pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

### 3.2 Studi Literatur

Studi Literatur adalah tahap awal yang digunakan untuk menambah studi pustaka dan pengetahuan yang mendukung untuk mengerjakan penulisan laporan dan penelitian. Studi literatur dilaksanakan dengan cara mengumpulkan teori dan pustaka yang berkaitan dengan penelitian ini meliputi:

1. Mekanisme hubungan sensor dengan mikrokontroler.
2. Pemrograman Arduino.

### 3.3 Analisa Kebutuhan

Pada tahap analisis kebutuhan dibagi menjadi dua bagian, yaitu kebutuhan fungsional dan kebutuhan non fungsional.

Pada bagian analisis kebutuhan *user* didalamnya hanya menjelaskan kebutuhan secara fungsional. Terdapat beberapa poin yang menjelaskan hal yang dapat *user* lakukan pada system yang menjadi kebutuhan *user* terhadap system yang dibangun. Hal tersebut berkaitan dengan fitur yang terdapat pada system. Kebutuhan *user* dapat di definisikan pada gambar di atas. Pada tahap analisis kebutuhan sistem, terdapat kebutuhan secara *functional* dan *non functional*. Kebutuhan *functional* pada sistem ini menjelaskan tentang hal yang dibutuhkan pada sistem agar sistem yang dibangun dapat berjalan dengan baik. Sedangkan kebutuhan *non functional* pada sistem ini menjelaskan tentang hal yang menjadi batasan terhadap kebutuhan perancangan sistem. Adapun kebutuhan sistem dapat didefinisikan menjadi dua yakni Kebutuhan Fungsional dan Kebutuhan Non-Fungsional :

Kebutuhan Fungsional merupakan kebutuhan yang harus ada pada sistem. Berikut merupakan kebutuhan fungsionalitas.

1. Sistem dapat mengetahui berat *silhouette*, dan getaran pada *silhouette*.
2. Sistem dapat melakukan perhitungan *K-Nearest Neighbor* sesuai dengan input dari *load cell* dan sensor getar.
3. Sistem dapat menerapkan aturan *K-Nearest Neighbor*.

Sistem dapat menghasilkan output berupa nilai skor sesuai dengan perhitungan *K-Nearest Neighbor*.

Dan Kebutuhan non-fungsional menjelaskan kebutuhan apa saja yang dibutuhkan oleh sistem terkait dengan kebutuhan perangkat keras maupun perangkat lunak.

#### 3.3.1 Kebutuhan Perangkat Keras

Untuk implementasi *scoring system* otomatis pada lomba menembak dengan target *silhouette* hewan, perangkat keras yang dibutuhkan adalah sebagai berikut:

1. Mikrokontroler Arduino Mega 2560 Kontroler Input (1)  
Digunakan untuk membaca data pada sensor getar, *load cell* dan *limit switch* yang kemudian dikirimkan ke pusat kontroler.
2. Mikrokontroler Arduino Mega 2560 Kontroler Pusat (2)  
Digunakan untuk mengolah data dari kontroler input.
3. Sensor Getar  
Digunakan untuk mendapatkan data getaran pada *silhouette*.
4. *Load Cell*  
Digunakan untuk mendapatkan data berat pada *silhouette*.

5. *Limit switch*

Digunakan sebagai trigger untuk mengirimkan data input sensor ke pusat kontroler.

6. Power 5V

Digunakan sebagai power dari kontroler.

7. PC/Laptop

Digunakan sebagai penampil score pada layar monitor.

### 3.3.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan *scoring system* otomatis pada lomba menembak dengan target *silhouette* hewan menggunakan metode *K-Nearest Neighbor (K-NN)* adalah sebagai berikut:

1. Arduino IDE

Digunakan sebagai tempat untuk melakukan pemrograman pada sistem.

2. *K-Nearest Neighbor*

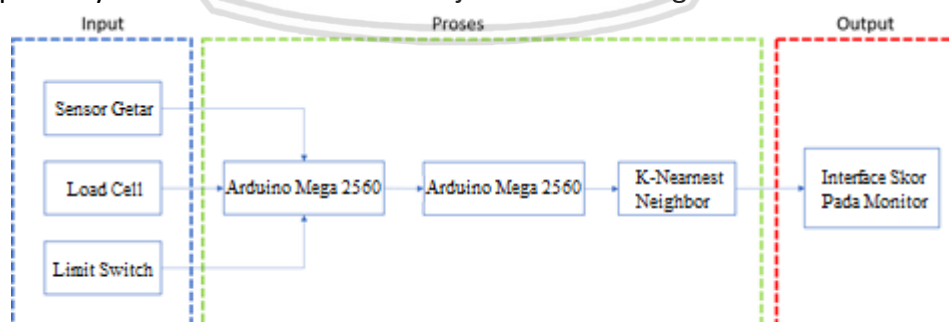
Digunakan sebagai penentu keputusan output.

3. Delphi 7

Digunakan sebagai tempat untuk membuat dan memprogram interface/antarmuka skor pada layar monitor.

### 3.4 Perancangan Sistem

Analisis kebutuhan yang untuk menghasilkan berbagai macam *software* dan *hardware* yang dibutuhkan untuk penelitian sehingga perancangan sistem dapat dibuat. Rancangan sistem terdiri dari beberapa blok yang ditunjukkan Pada Gambar 3.4. pada blok diagram tersebut terdapat 3 blok input yaitu sensor getar, *load cell* dan *limit switch*. Mikrokontroler yang digunakan adalah Arduino Mega 2560(1) yang merupakan kontroler untuk membaca input yang kemudian mengirimkan ke pusat kontroler dan Arduino Mega 2560 (2) yang digunakan sebagai pusat kontroler. Arduino Mega 2560(2) mengolah data dengan menggunakan *K-Nearest Neighbor* untuk memberikan output berupa interface skor pada layar monitor. Berikut akan dijelaskan melalui gambar 3.3 berikut.



**Gambar 3.3** Diagram Blok Rancangan Sistem

### 3.5 Implementasi

Tahap implementasi sistem akan dilakukan berdasarkan perancangan yang telah ditentukan sebelumnya. Pada tahap ini akan diimplementasikan seluruh gagasan serta ide baik desain maupun secara perhitungan sistem yang telah menjadi target tujuan sebelumnya. Terdapat dua tahap implementasi yaitu Konfigurasi Perangkat Keras dan Konfigurasi Perangkat Lunak.

#### 3.5.1 Konfigurasi Perangkat Keras

Pada tahap ini, yang harus dilakukan adalah dengan merancang *load cell*, sensor getar dan *limit switch* untuk dihubungkan dengan kontroler input (Arduino Mega 2560) dan kemudian kontroler input (Arduino Mega 2560(1)) dihubungkan ke pusat kontroler (Arduino Mega 2560 (2)). Sensor-sensor tersebut dihubungkan menggunakan kabel jumper. Konfigurasi perangkat keras dilakukan berdasarkan datasheet pada masing-masing sensor yang digunakan, Arduino Mega 2560(1) Kontroler input dan Arduino Mega(2) Kontroler pusat.

#### 3.5.2 Konfigurasi Perangkat Lunak

Pada tahap konfigurasi perangkat lunak, dilakukan penginstallan *software* Arduino IDE, kemudian dilakukan coding program dengan menggunakan Bahasa C yang bertujuan untuk mengkonfigurasi jalannya sistem scoring pada lomba menembak dan *K-Nearest Neighbor (K-NN)* untuk menentukan output berdasarkan dari hasil perhitungannya.

### 3.6 Pengujian dan Analisis

Pengujian akan dilakukan kepada keseluruhan sistem yaitu dengan menampilkan hasil pengujian terhadap output yang diberikan. Output diberikan melalui perhitungan *K-Nearest Neighbor (K-NN)* berdasarkan data yang telah diterima pada sensor yang kemudian dikirimkan ke pusat kontroler (Arduino Mega(2)). Terdapat 4 pengujian yang akan dilakukan, yaitu pengujian akuisisi data sensor getar, akuisisi data *load cell*, akuisisi data *limit switch*, dan proses *K-Nearest Neighbor (K-NN)*. Data yang didapatkan dari hasil pengujian akan ditampilkan dalam bentuk tabel maupun grafik.

### 3.7 Kesimpulan

Kesimpulan didapatkan setelah melakukan tahap perancangan, implementasi, pengujian dan analisis terhadap sistem yang dibuat. Kesimpulan disusun berdasarkan hasil dari tahap pengujian dan analisis yang dilakukan pada sistem yang dibuat. Isi pada kesimpulan diharapkan dapat menjadi acuan dasar pada penelitian selanjutnya untuk mengembangkan *scoring system* otomatis pada lomba menembak. Pada akhir penulisan terdapat saran yang bertujuan untuk memberikan kemudahan penelitian selanjutnya, apabila akan meneruskan dan mengembangkan penelitian ini.



## BAB 4 REKAYASA KEBUTUHAN

### 4.1 Gambaran Umum Sistem

*Scoring* sistem otomatis pada lomba menembak dengan target *silhouette* hewan menggunakan metode *K-NN (K-Nearest Neighbor)* merupakan sebuah sistem yang dapat menentukan apakah target tersebut masuk kedalam kriteria skoring atau bukan. Sistem membutuhkan keadaan awal yang dimana target *silhouette* belum diletakan di atas tempat yang disediakan untuk target *silhouette*. Kemudian sistem akan mengambil data sensor yang kemudian dikirimkan ke pusat kontroler. Kontroler akan mengambil data pada sensor getar, *load cell*, dan *limit switch* yang kemudian dikirimkan ke pusat kontroler. Metode *K-NN (K-Nearest Neighbor)* digunakan untuk menentukan target tembak dan skor yang akan ditampilkan.

### 4.2 Analisa Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan agar keseluruhan kebutuhan yang diperlukan untuk membangun *scoring* sistem otomatis pada perlombaan menembak dengan target *silhouette* hewan ini dapat dipenuhi. Dalam melakukan analisis kebutuhan sistem ini, akan dibagi menjadi 2 bagian yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

#### 4.2.1 Kebutuhan Fungsional

Kebutuhan Fungsional merupakan kebutuhan dari sistem yang harus di penuhi yaitu :

1. Sistem dapat mengetahui berat *silhouette*, dan getaran pada *silhouette*.

a. Penjelasan dan Prioritas

Untuk mendapatkan input nilai pada sensor getar, *load cell* maupun pada *limit switch*. Sensor getar berfungsi untuk mendeteksi getaran pada saat *silhouette* terjatuh. *Load cell* berfungsi untuk mengukur berat pada masing-masing *silhouette*. *Limit switch* berfungsi sebagai trigger untuk melakukan pengiriman data dari kontroler input ke pusat kontroler.

b. Stimulus/respon sistem

Pada saat kontroler input dan pusat kontroler terhubung dengan daya, maka sistem akan berjalan sesuai dengan prioritas pertama yaitu pembacaan sensor yang kemudian akan melanjutkan ke proses selanjutnya. Respon ini tidak bisa dilihat oleh *user*, namun proses ini dapat dilihat pada serial monitor pada Arduino IDE sehingga dapat melihat data pada masing-masing sensor.

c. Kebutuhan Fungsional

Fungsi ini harus terpenuhi oleh sistem dikarenakan menjadi input pada sistem. Nilai input dari sistem ini akan dikomputasikan oleh Arduino Mega 2560(2) (Pusat Kontroler) .

2. Sistem dapat melakukan perhitungan *K-NN (K-Nearest Neighbor)* sesuai dengan input dari sensor getar, dan *load cell*.
  - a. Penjelasan dan Prioritas  
Fungsi ini berguna untuk sistem agar dapat melakukan perhitungan *K-NN (K-Nearest Neighbor)* mulai dari menentukan parameter K (jumlah tetangga paling dekat) untuk menghitung kuadrat jarak eucliden objek terhadap data training yang diberikan, setelah itu diurutkan berdasarkan objek-objek yang mempunyai jarak Euclidian dari nilai yang tertinggi ke yang terendah yang nantinya digunakan untuk mendapatkan kategori Y dan digunakan sebagai klasifikasi *K-NN (K-Nearest Neighbor)* merupakan bagian output dari sistem.
  - b. Stimulus/respon sistem  
Ketika sistem mendapatkan input pada kedua sensor, maka sistem akan menentukan parameter K (jumlah tetangga paling dekat) yg sudah di inputkan tadi untuk menghitung kuadrat jarak eucliden objek terhadap data training yang diberikan, kemudian akan diurutkan berdasarkan objek-objek yang mempunyai jarak Euclidian dari nilai yang tertinggi ke yang terendah yang nantinya digunakan untuk mendapatkan kategori Y dan digunakan sebagai klasifikasi *K-NN (K-Nearest Neighbor)* merupakan bagian output dari sistem.
  - c. Kebutuhan Fungsional  
Fungsi ini berperan penting agar perhitungan *K-NN (K-Nearest Neighbor)* dapat menghasilkan output yang sesuai. Apabila fungsi ini tidak berjalan maka sistem tidak akan menghasilkan output yang akurat sesuai dengan perhitungan *K-NN (K-Nearest Neighbor)*.
3. Sistem dapat menerapkan aturan *K-NN (K-Nearest Neighbor)*.
  - a. Penjelasan dan Prioritas  
Fungsi ini mengharuskan sistem agar bisa menerapkan aturan - aturan dan sistem dapat membedakan jenis *silhouette*, sehingga dapat digunakan untuk inferensi sebagai salah satu tahap untuk mendapatkan ouput dengan menggunakan metode *K-NN (K-Nearest Neighbor)* yang kemudian digunakan untuk proses pengklasifikasian.
  - b. Stimulus/respon sistem  
Pada saat sistem telah melakukan perhitungan untuk mendapatkan nilai mulai dari menentukan parameter K (jumlah tetangga paling dekat) untuk menghitung kuadrat jarak eucliden objek terhadap data training yang diberikan, setelah itu diurutkan berdasarkan objek-objek yang mempunyai jarak Euclidian dari nilai

yang tertinggi ke yang terendah yang nantinya digunakan untuk mendapatkan kategori Y dan digunakan sebagai klasifikasi *K-NN (K-Nearest Neighbor)*. Setelah rule ditentukan sesuai dengan urutan perhitungan *K-NN (K-Nearest Neighbor)*, maka sistem akan mendapatkan hasil dari pengklasifikasian dari data yang diinputkan.

c. Kebutuhan Fungsional

Fungsi ini harus bisa dipenuhi oleh sistem karena output sistem bergantung pada rule yang dibuat. Rule yang dimaksud adalah ketika sistem memasuki "mode ayam", maka penambahan nilai keluaran hanya dapat terjadi ketika *silhouette* ayam yang tertembak. Hal yang sama terjadi ketika sistem memasuki mode dengan *silhouette* masing-masing. Rule yang telah dibuat ini harus lengkap sesuai dengan himpunan *K-NN (K-Nearest Neighbor)* yang ada, karena jika salah satu rule tidak lengkap maka output yang dikeluarkan oleh sistem tidak sesuai.

4. Sistem dapat menghasilkan output berupa nilai skor sesuai dengan perhitungan *K-NN (K-Nearest Neighbor)*.

a. Penjelasan dan Prioritas

Fungsi ini mengharuskan sistem agar bisa memberikan kondisi output pada layar monitor agar *user* lebih mudah mengamati nilai skor.

b. Stimulus/respon sistem

Pada saat seluruh proses *K-NN (K-Nearest Neighbor)* telah dilakukan dan sistem dapat menentukan outputnya, maka monitor akan mengeluarkan nilai skor.

c. Kebutuhan Fungsional

Fungsi ini berguna untuk memberikan output dari sistem sebagai skoring yang akan tampil pada layar monitor.

#### 4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional pada sistem akan dibagi menjadi 2 kebutuhan, yaitu kebutuhan perangkat keras dan perangkat lunak.

Perangkat keras dibutuhkan dalam membuat prototipe dari system ini, diantaranya adalah :

1) Menghitung Getar

Getaran merupakan suatu gerak bolak-balik pada sekitar kesetimbangan. Untuk menghitung Getaran pada *Silhouette* digunakan sensor Getar yakni sensor getar SW-420 yang dimana untuk mengukur getaran *silhouette* ketika jatuh. Sensor ini diletakkan di bagian belakang *silhouette*. Saat *silhouette* bergetar maka keluaran hasil data pada sensor getar akan di masukkan sebagai *input* dari proses *K-NN (K-Nearest Neighbor)*. Hasil

keluaran dari sensor SW-420 merupakan nilai *Strength of Vibration* seperti amplitudo dari gelombang getaran yang dihasilkan.

2) Menghitung Berat

Untuk menghitung Berat pada *Silhouette* digunakan sensor *Load Cell* yakni sensor yang mengukur kondisi berat pada *silhouette*. Sensor ini di letakkan di bawah *silhouette* yang dimana ketika jatuh berat yang diambil menjadi perhitungan berat yang nantinya gunakan sebagai input dari proses K-NN (*K-Nearest Neighbor*).

3) Melakukan Klasifikasi *K-NN (K-Nearest Neighbor)* dengan menggunakan dua mikrokontroler Arduino Mega 2560

Saat data perhitungan hasil dari Berat dan Getar dari tiap *silhouette* sudah di dapatkan maka klasifikasi K-NN akan didapatkan, Data Training yang di gunakan sebagai acuan hasil valid akan di ujikan dengan data uji atau acak yang dimana data uji tersebut akan di klasifikasi dengan Data Training yang sudah ada apakah Data yang di Uji tersebut menghasilkan output yang sesuai atau tidak. Digunakan dua Arduino Mega 2560 karena setiap Arduino Mega hanya mampu melakukan proses sensor setiap dua *silhouette* saja.

#### 4.3 Batasan Desain Sistem

Dalam pembuatan prototype dari *scoring* sistem otomatis pada perlombaan menembak dengan target *silhouette* hewan terdapat Batasan yang diterapkan sehingga dalam proses pembahasan, perancangan dan implementasi dari prototype sistem ini tidak mencakup jangkauan yang luas. Batasan-batasan dari desain prototype sistem tersebut antara lain adalah :

1. Sistem memperoleh input berat ketika *silhouette* belum diletakan pada sasaran menembak untuk kemudian dijadikan sebagai berat awal.
2. Sistem melakukan perhitungan jumlah *silhouette* yang dijadikan sebagai acuan input dari prototipe system.
3. Sistem melakukan konfirmasi apakah *Silhouette* yang jatuh merupakan yang ditargetkan atau bukan.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

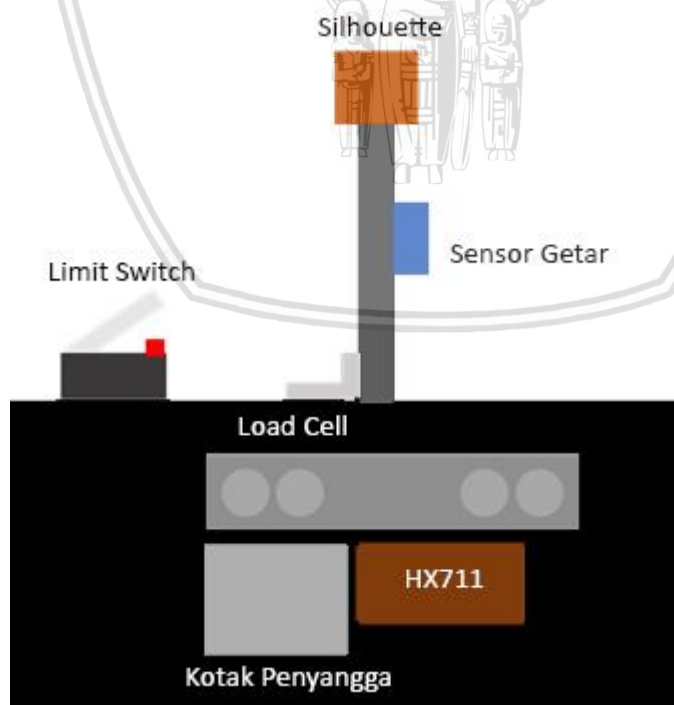
Pada bab ini akan menjelaskan proses perancangan dan implementasi dari “*Scoring System Otomatis pada Perlombaan menembak dengan Target Silhouette Hewan Dengan Menggunakan Metode K-Nearest Neighbor (K-NN)*”. Terdapat dua perancangan yaitu perancangan perangkat lunak (*Software*) dan perancangan perangkat keras (*Hardware*). Sedangkan pada implementasi terdapat implementasi perangkat lunak (*Software*) dan implementasi perangkat keras (*Hardware*).

### 5.1 Perancangan Sistem

Pada perangkat yang dirancang, sistem dapat melakukan *scoring* berdasarkan dari berat pada *silhouette*, dan getaran pada *silhouette* ketika ditembak. Input berupa berat dan getaran pada *silhouette* yang diolah menggunakan mikrokontroler. Output dari sistem menggunakan metode *K-Nearest Neighbor* untuk menentukan apakah *silhouette* yang jatuh termasuk dalam mode penembakan.

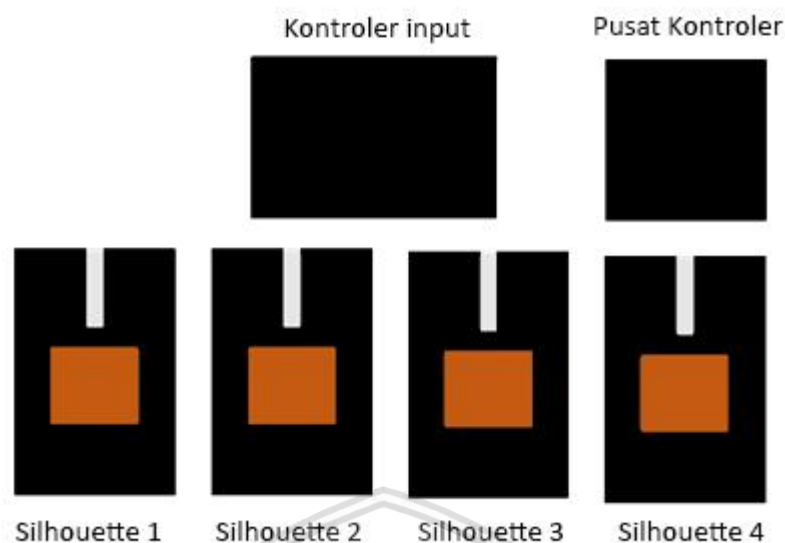
#### 5.1.1 Perancangan Alat Mekanik, Peletakan dan Silhouette

Dalam perancangan alat mekanik, peletakan dan *silhouette* menggunakan sensor getar, *load cell*, *limit switch* dan modul *HX711* yang diletakan pada box kotak yang berbahan plastik.



Gambar 5.1 Perancangan Alat Tampak Samping





**Gambar 5.2** Perancangan Alat Tampak Atas

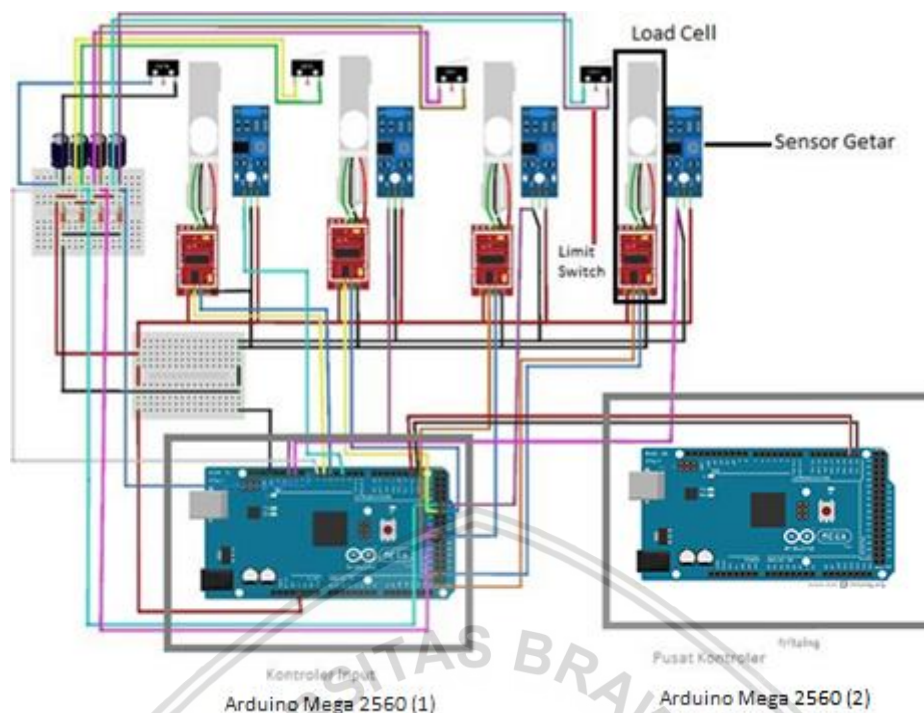
Pada Gambar 5.1 dijelaskan bahwa bagian atas box *silhouette* terdapat *limit switch*, sensor getar serta *silhouettenya*, sedangkan bagian dalam box, terdapat *load cell*, modul *HX711*.

Pada Gambar 5.2 dijelaskan bahwa jarak antara kontroler input dan pusat kontroler berjarak 5,5cm, sedangkan jarak antara *silhouette1*, *silhouette2*, *silhouette3*, dan *silhouette4* berjarak 3,5cm. Gambar 5.2 merupakan gambar alat tampak atas.

### 5.1.2 Perancangan Perangkat Keras

Perangkat keras (*Hardware*) digunakan untuk membangun sistem *scoring* yang mendukung mikrokontroler untuk menerapkan Metode *K-Nearest Neighbor (K-NN)* sebagai pengambil keputusan output. Mikrokontroler yang digunakan adalah Arduino Mega 2560. Input yang digunakan untuk mengukur berat *silhouette* adalah *load cell* untuk mengukur berat dan sensor getar untuk mengukur getaran pada *silhouette*.

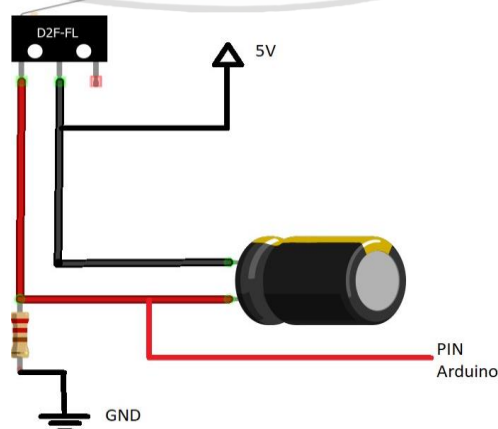




**Gambar 5.3** Skema Perancangan Perangkat Keras

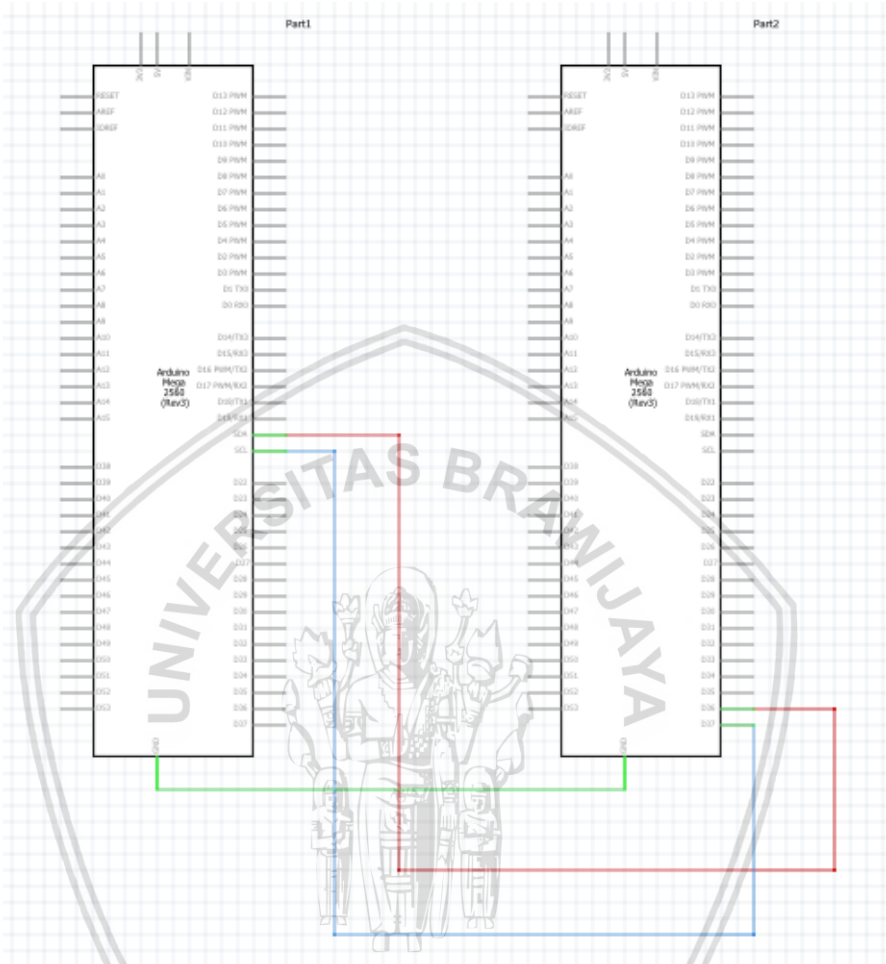
Pada blok diagram diatas, input data berupa sensor getar, *load cell*, dan *push button*. Selanjutnya data pada sensor getar, *load cell*, dan *push button* yang telah diambil akan di proses pada Arduino sehingga sistem mengetahui skor dan jenis *silhouette* yang tertembak. Kemudian hasil dari proses yang dilakukan pada Arduino akan ditampilkan melalui monitor.

Pada *limit switch* terdapat rangkaian debounce untuk menghilangkan efek bouncing yang terjadi pada *limit switch*, karena efek bouncing ini akan menyebabkan nilai dari variabel *limit switch* tidak akurat, ketika ditekan *limit switch* value nya tidak tentu antara 0 atau 1. Berikut merupakan gambar rangkaian debounce yang dihubungkan pada *limit switch*



**Gambar 5.4** Rangkaian Debounce

Untuk komunikasi antar Dua buah Arduino Mega menggunakan I2C. Arduino Mega pertama digunakan sebagai *master writer* dan arduino mega kedua sebagai *slave receiver*. Berikut merupakan gambar dari koneksi I2C antara Kedua Arduino Mega.



**Gambar 5.5** Koneksi I2C Antar Arduino

Koneksi pin Gambar 5.3 ditunjukkan pada Tabel 5.1 berikut

**Tabel 5.1** Koneksi Pin Perancangan Perangkat Keras

Pin Arduino Mega 1	Pin Arduino Mega 2	Pin Load Cell	Pin Sensor Getar	Pin Limit Switch
Vcc	-	Vcc	Vcc	Vcc
GND	GND	GND	GND	GND
SDA	SDA 20	-	-	-
SCL	SDA 21	-	-	-
3	-	-	Getar1	-

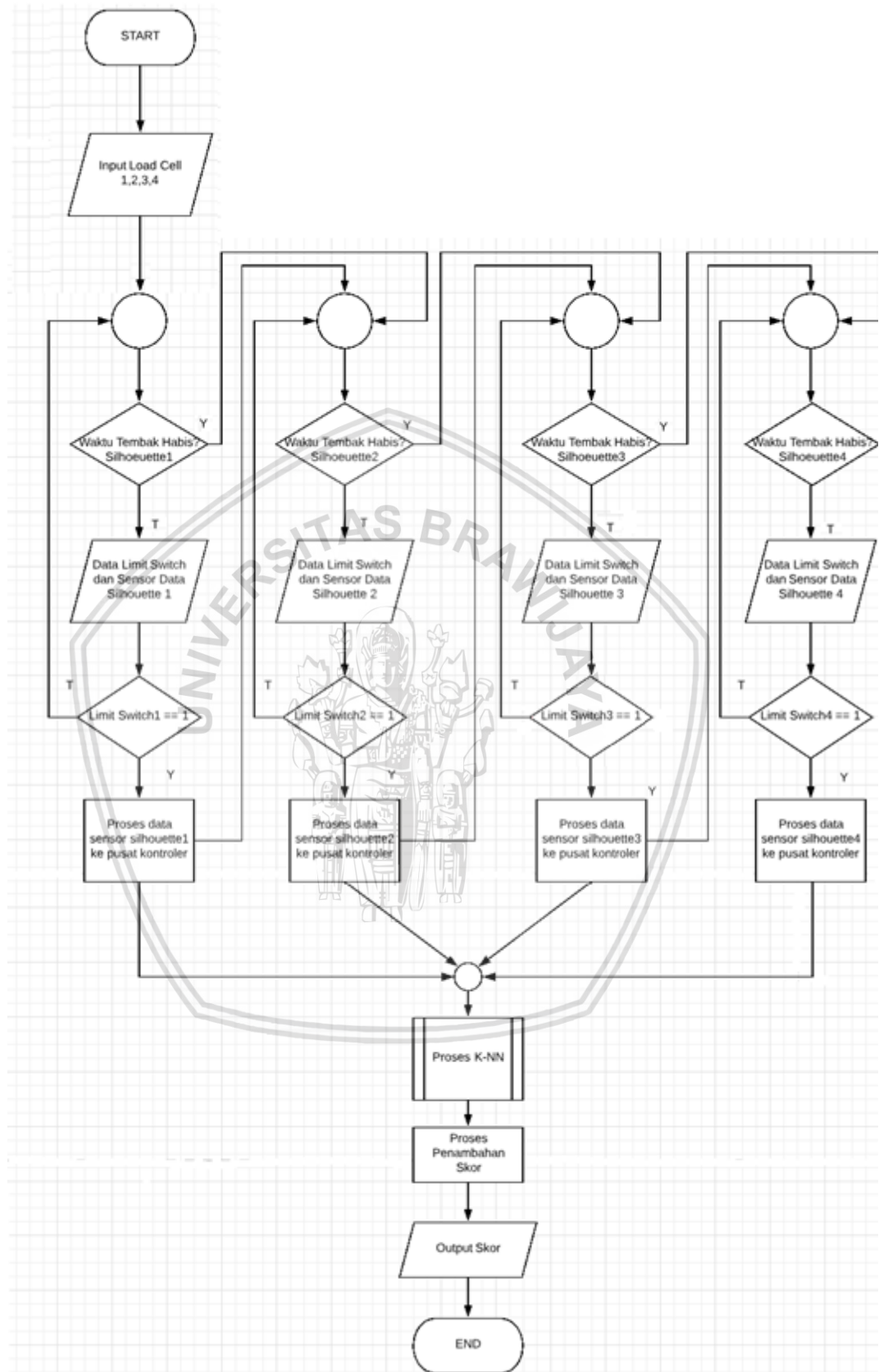
Pin Arduino Mega 1	Pin Arduino Mega 2	Pin Load Cell	Pin Sensor Getar	Pin Limit Switch
4	-	-	-	Limit1
5	-	SCK1	-	-
6	-	DOUT1	-	-
9	-	-	-	Limit4
10	-	-	Getar4	-
11	-	-	Getar3	-
28	-	-	-	Limit2
31	-	-	Getar2	-
32	-	SCK2	-	-
33	-	DOUT2	-	-
39	-	-	-	Limit3
40	-	SCK3	-	-
41	-	DOUT3	-	-
52	-	DOUT4	-	-
53	-	SCK4	-	-

### 5.1.3 Perancangan Perangkat Lunak

Perangkat lunak pada sistem ini berupa code program yang digunakan untuk mikrokontroller yang bernama Arduino IDE. Code program yang dibuat dengan menerapkan metode *K-Nearest Neighbor (K-NN)* sebagai pengambil keputusan untuk output berupa skor yang berdasarkan input dari *load cell* yang berupa berat *silhouette*, dan sensor getar untuk mengetahui getaran pada *silhouette*.

Implementasi *K-Nearest Neighbor (K-NN)* ke dalam *scoring* sistem otomatis pada perlombaan menembak, akan dibuat suatu variabel array yang berfungsi sebagai tempat data dari data training yang sudah ada sebelumnya supaya penyesuaian data yang akan di ujikan akurat dan sesuai dengan perhitungan *K-Nearest Neighbor (K-NN)*.

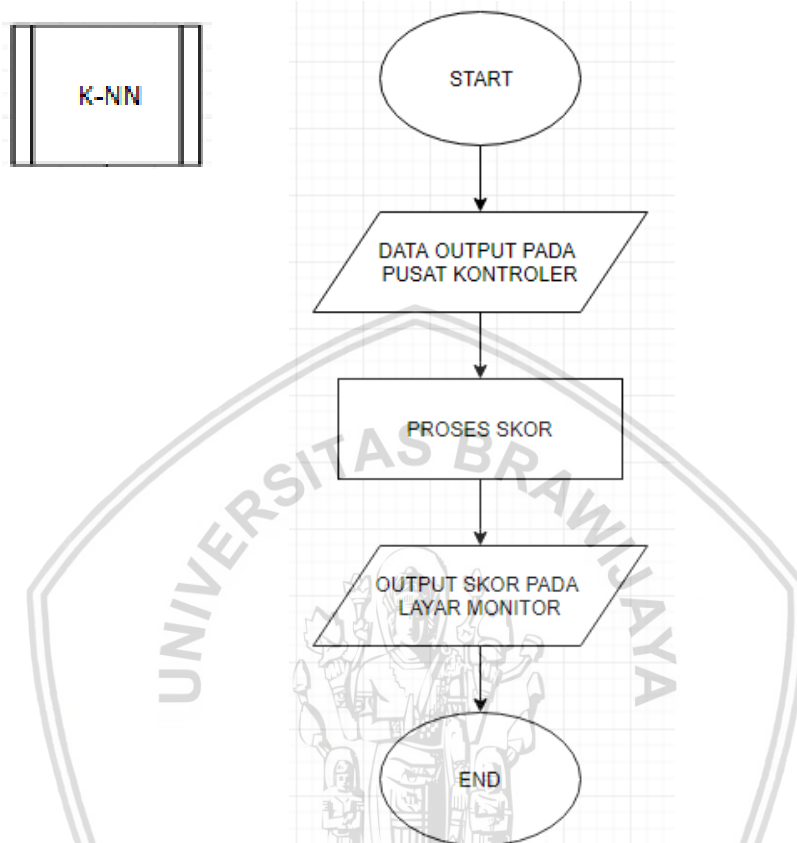
Output pada *scoring* sistem otomatis pada perlombaan menembak berupa skor yang ditampilkan pada layer monitor yang merupakan hasil dari perhitungan *K-Nearest Neighbor (K-NN)*.



**Gambar 5.6** Flowchart Perancangan Perangkat Lunak

#### 5.1.4 Perancangan Papan Skor

Pada perancangan papan skor, sistem memiliki fungsi untuk menampilkan skor pada layar monitor berdasarkan hasil output yang dikeluarkan sistem yang akan dijelaskan pada Gambar 5.7 berikut.



Gambar 5.7 Flowchart Perancangan Papan Skor

#### 5.1.5 Perancangan *K-Nearest Neighbor (K-NN)*

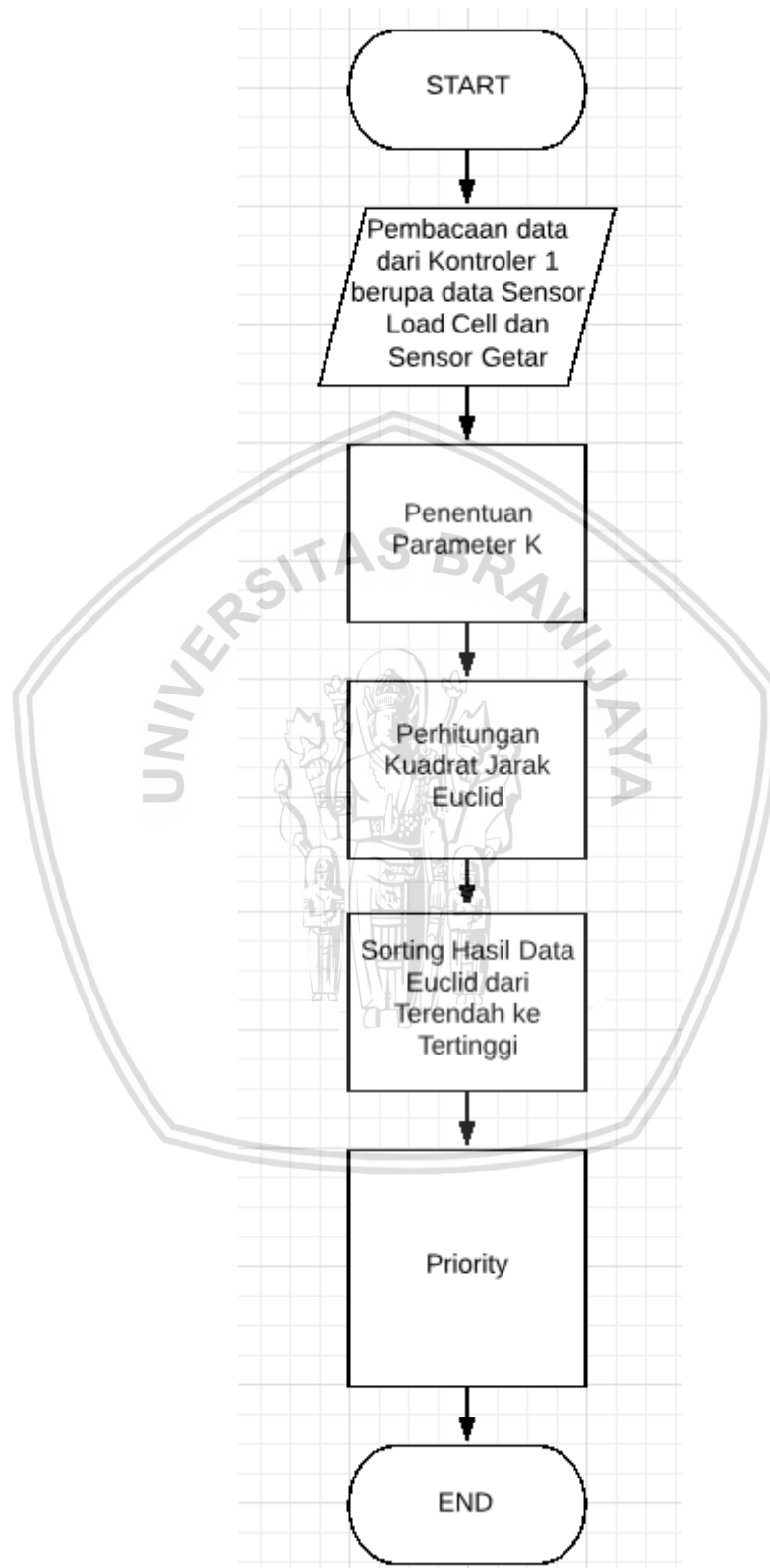
Perancangan sistem menggunakan metode *K-Nearest Neighbor (K-NN)* membutuhkan beberapa proses sehingga didapatkannya suatu keputusan output dari sistem sesuai dengan perhitungan *K-Nearest Neighbor (K-NN)*. Proses pada metode *K-Nearest Neighbor (K-NN)* diantaranya :

- Menentukan Parameter K ( Jumlah tetangga paling dekat ) Dengan Melakukan testing pada data training dan nilai K dari data yang coba diujikan. Dengan Rumus sbg berikut :

$$[\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}] =$$

- Menghitung Kuadrat jarak Euclid . Setelah data di masukkan ke masing – masing variable yakni parameter  $\sum$  Data Training
- Mengurutkan Objek-objek yang memiliki nilai terkecil hingga nilai terbesar.
- Hasil Output berupa Nilai Kelas dari klasifikasi *K-Nearest Neighbor*.

Pada Gambar 5.6 menunjukkan sub-proses pada *K-Nearest Neighbor (K-NN)* yang memiliki fungsi saling berhubungan dengan sub-proses lainnya.



**Gambar 5.8** Flowchart Perancangan *K-Nearest Neighbor (K-NN)*



## 5.2 Implementasi Sistem

Pada sub-bab implementasi sistem, akan dijelaskan tentang system pada perangkat keras maupun perangkat lunak.

### 5.2.1 Implementasi Alat Mekanik, Peletakan dan *Silhouette*

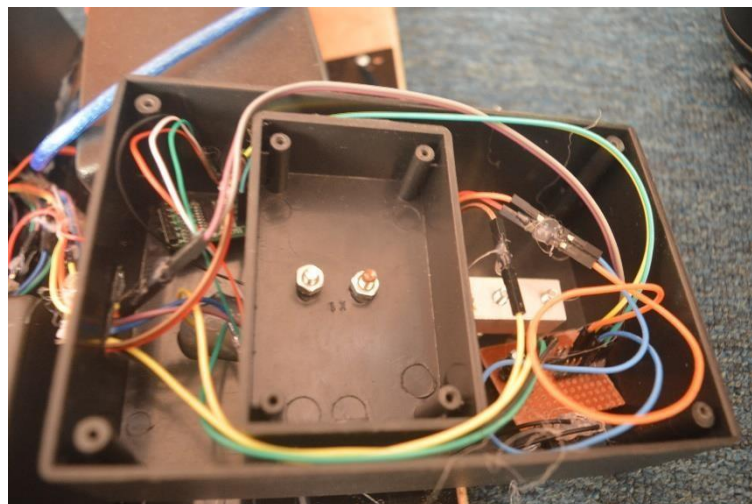
Pada tahap implementasi alat mekanik, peletakan, dan *silhouette*, telah disebutkan sebelumnya pada sub bab 5.1.1 yaitu perancangan alat mekanik, peletakan dan *silhouette*. Alat yang digunakan sebagai box *silhouette* berbahan plastik, dan alas yang digunakan sebagai dasar papan menggunakan bahan akrilik dan kayu yang telah dipotong sesuai keperluan.



Gambar 5.9 Implementasi Prototipe Tampak Depan

### 5.2.2 Implementasi Perangkat Keras

Dalam melakukan tahap implementasi perangkat keras harus disesuaikan pada perancangan perangkat keras yang telah di buat pada sub bab 5.1.2. Sistem menggunakan sensor getar, *load cell*, dan *limit switch*. *Load cell* digunakan untuk mengukur berat pada *silhouette*, sensor getar digunakan untuk mengukur getaran pada *silhouette* ketika jatuh, dan *limit switch* digunakan sebagai trigger untuk mengirimkan data *silhouette* ke pusat kontroler. Berikut adalah hasil dari implementasi perangkat keras Pada Gambar 5.10 .



**Gambar 5.10** Implementasi Rangkaian Pada Box *Silhouette*

Pada Gambar 5.10 merupakan implementasi rangkaian pada box *silhouette* yang dibuat dari bahan plastik. Didalam kotak tersebut terdapat *load cell* yang berguna untuk mengukur berat dari *silhouette*, dan pada bagian atas kotak terdapat *limit switch* dan switch yang nantinya juga merupakan implementasi rangkaian pada box *silhouette* yang digunakan untuk membaca data sensor yang dihasilkan pada masing-masing box *silhouette* lalu diolah data sensor tersebut dengan menggunakan Metode *K-Nearest Neighbor (K-NN)*.

### 5.2.3 Implementasi Perangkat Lunak

Pada sub-bab implementasi perangkat lunak, terdapat beberapa bagian subprogram yaitu program pembacaan nilai pada setiap sensor, pengiriman data antara kontroler, tampilan output pada monitor, penentuan parameter K, Menghitung kuadrat jarak euclid, pengurutan objek-objek data mulai dari terendah sampai tertinggi, dan Priority. Untuk mendapatkan hasil output pada system. Program menggunakan Bahasa C pada Arduino IDE dan Delphi.

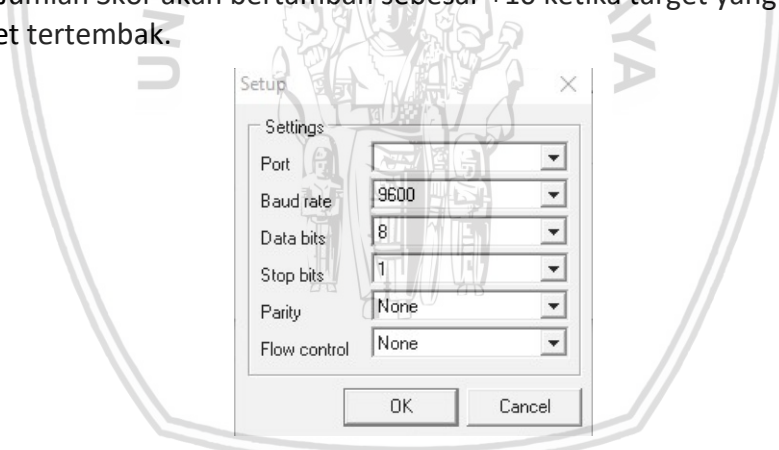
### 5.2.4 Implementasi Papan Skor

Pada tahap implementasi ini, papan skor yang dimaksud adalah antar muka untuk nilai skor pada perlombaan menembak yang akan ditampilkan pada layar monitor. Berikut merupakan tampilan interface papan skor.



**Gambar 5.11** Papan Skor pada Layar Monitor

Pada Gambar 5.11 merupakan tampilan papan skor yang akan tampil pada layar monitor. Pada tampilan tersebut *user* disediakan menu “*Setting*”, “*Connect*”, “*Disconnect*”, dan “*Close*”. *User* juga dapat melakukan koneksi maupun memutuskan koneksi pada kontroler. Untuk *setting* digunakan untuk memilih dan mengatur sambungan dari *port* Arduino IDE yang digunakan. *Connect* digunakan untuk menyambungkan *port* yang sudah di pilih di menu *setting*. *Disconnect* digunakan untuk memutuskan sambungan *port* dari Arduino IDE. *Close* digunakan untuk menutup program. Skor Pemain merupakan tampilan hasil skor yang di dapatkan pemain sedangkan Jenis Target adalah target yang tertembak oleh pemain. Jumlah Skor akan bertambah sebesar +10 ketika target yang dituju benar dan target tertembak.



**Gambar 5.12** Menu *Setting*

Pada Gambar 5.12 merupakan tampilan menu setting yang digunakan untuk melakukan konfigurasi awal pada sistem. Pada tampilan tersebut *user* dapat mengatur *port* yang digunakan pada kontroler. Untuk menu *Port* yang di pilih yakni sesuai dengan *port* yang ada dalam Arduino IDE. Menu *Baud rate* digunakan untuk memilih berapa jumlah kali per detik sinyal dalam perubahan data komunikasi analog yang di inginkan *user*. Menu Data Bits digunakan untuk memilih berapa bit yang di butuhkan oleh *user* untuk program tersebut. Menu Stop bits yakni digunakan mengawali dan mengakhiri paket data dalam komunikasi serial , terdapat 1 ; 1,5 ; 2 stop bit. Menu *Parity* digunakan sebagai checker apakah bit yang di dikirimkan itu berupa nilai ganjil atau genap. Dan Flow control adalah mekanisme untuk pengirim dan penerima untuk mengontrol kecepatan transmisi dari software , hardware ataupun custom.

### 5.2.5 Implementasi *K-Nearest Neighbor (K-NN)* pada Sistem *Scoring*

Untuk mengimplementasikan *K-Nearest Neighbor (K-NN)*, pertama kali yang dilakukan untuk memprogram Arduino pada Arduino IDE dengan menginisialisasi library apa saja yang digunakan, menginisialisasi pin-pin yang digunakan pada sensor ke Arduino, dan void setup. Untuk menerapkan Metode *K-Nearest Neighbor (K-NN)* ke dalam sistem *scoring*, pertama kali yang dilakukan oleh sistem dalam membaca input dari *load cell*, sensor getar, dan *limit switch*. Dari hasil input tersebut akan dibuat perhitungan sistem *K-Nearest Neighbor (K-NN)* pada variabel berat, dan getaran. Untuk perhitungan sistem *K-Nearest Neighbor (K-NN)* adalah sebagai berikut.

**Tabel 5.2** Code Program Klasifikasi *K-Nearest Neighbor*

No	Code Program
1	<code>//#include &lt;ArduinoSort.h&gt;</code>
2	
3	<code>//String kelas[] = {"Ayam Normal", "Ayam Tersentuh", "Ayam</code>
4	<code>Tertembak", "Kalkun Normal", "Kalkun Tersentuh", "Kalkun</code>
5	<code>Tertembak", "Babi Normal", "Babi Tersentuh", "Babi</code>
6	<code>Tertembak", "Domba Normal", "Domba Tersentuh", "Domba</code>
7	<code>Tertembak"};</code>
8	<code>String kelas[] = {"Ayam Normal",</code>
10	<code>"Ayam Tersentuh" ,</code>
11	<code>"Ayam Tertembak" ,</code>
12	<code>"Kalkun Normal",</code>
13	<code>"Kalkun Tersentuh",</code>
14	<code>"Kalkun Tertembak",</code>
15	<code>"Babi Normal",</code>
16	<code>"Babi Tersentuh",</code>
17	<code>"Babi Tertembak",</code>
18	<code>"Domba Normal",</code>
19	<code>"Domba Tersentuh",</code>
20	<code>"Domba Tertembak"</code>
21	<code>};</code>
22	<code>const size_t totalDataLatih = 120;</code>
23	
24	<code>float datalatih[totalDataLatih][3] = {{33.02,932,0},</code>
25	<code>{37.8,4999,0},</code>
26	<code>{31.94,1840,0},</code>
27	<code>{32.67,5043,0},</code>
28	<code>{30.6,6183,0},</code>
29	<code>{28.04,9507,0},</code>
30	<code>{25.38,4424,0},</code>
31	<code>{29.99,4077,0},</code>
32	<code>{24.5,7647,0},</code>
33	<code>{27.47,8341,0},</code>
34	<code>{33.62,21268,1},</code>
35	<code>{34.13,19670,1},</code>
36	<code>{30.16,19118,1},</code>
37	<code>{32.98,17702,1},</code>

No	Code Program
38	{31.9,12560,1},
39	{28,15600,1},
40	{27.63,22160,1},
41	{33,23938,1},
42	{37.8,15879,1},
43	{31.92,17761,1},
44	{32.12,31775,2},
45	{30.32,31255,2},
46	{27.32,25647,2},
47	{26.77,21311,2},
48	{26.38,26102,2},
49	{27.45,26207,2},
50	{27.38,28161,2},
51	{27.8,25008,2},
52	{26.56,21080,2},
53	{29.1,28782,2},
54	{35.8,8313,3},
55	{36.86,12810,3},
56	{43.08,6676,3},
57	{36.15,5432,3},
58	{30.54,7091,3},
59	{36.21,6617,3},
60	{36.66,3192,3},
61	{36.83,6066,3},
62	{35.36,8992,3},
63	{33.99,7767,3},
64	{40.5,7240,4},
65	{37.02,20640,4},
66	{35.74,21292,4},
67	{35.01,15156,4},
68	{35.25,25484,4},
69	{35.8,16805,4},
70	{36.86,25082,4},
71	{43.08,11193,4},
72	{36.15,16202,4},
73	{32.14,15748,4},
74	{36.21,20595,5},
75	{36.66,30771,5},
76	{36.83,23048,5},
77	{35.36,31629,5},
78	{33.99,24468,5},
79	{33.19,24770,5},
80	{36.62,23100,5},
81	{35.43,24705,5},
82	{36.04,25169,5},
83	{37.02,27428,5},
84	{44.16,5101,6},
85	{45.33,6498,6},
86	{46.9,5226,6},



No	Code Program
90	{52.73,5969,6},
91	{47.27,5467,6},
92	{46.4,7633,6},
93	{49.38,8930,6},
94	{43.36,8046,6},
95	{47.27,5978,6},
96	{46.9,8431,6},
97	{44.9,19143,7},
98	{47.08,17091,7},
99	{47.27,21950,7},
100	{47.21,26320,7},
101	{46.43,8295,7},
102	{44.04,31020,7},
103	{46.4,18752,7},
104	{49.38,8107,7},
105	{43.36,11712,7},
106	{47.01,17367,7},
107	{44.16,22854,8},
108	{45.33,29172,8},
109	{46.9,28158,8},
110	{52.57,30135,8},
111	{47.79,28690,8},
112	{42.73,22291,8},
113	{48.29,24367,8},
114	{45.56,24035,8},
115	{47,29278,8},
116	{55,28390,8},
117	{74.9,3268,9},
118	{75,3100,9},
119	{59.95,3950,9},
120	{68.89,4730,9},
121	{68.49,1973,9},
122	{68.13,4193,9},
123	{64.39,3168,9},
124	{66.78,2228,9},
125	{64.4,5295,9},
126	{69.71,2334,9},
127	{62.65,28171,10},
128	{63.4,6174,10},
129	{53.2,7210,10},
130	{61.23,21913,10},
131	{67.21,9906,10},
132	{61.23,21913,10},
133	{63.21,10070,10},
134	{64.19,16456,10},
135	{68.12,7528,10},
136	{68.23,16251,10},
137	{64,24385,11},
138	{64.85,26282,11},
139	
140	
141	
142	

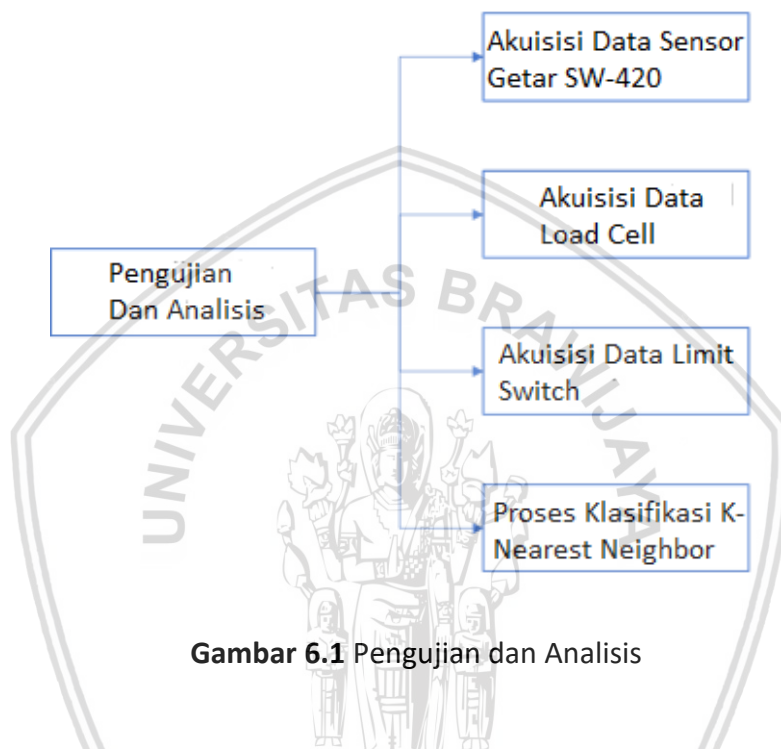


No	Code Program
143	<code>{70.09,31002,11},</code>
144	<code>{65.98,30418,11},</code>
145	<code>{74.2,29028,11},</code>
146	<code>{73.89,29526,11},</code>
147	<code>{57.45,27052,11},</code>
148	<code>{72.83,24697,11},</code>
149	<code>{67.22,25848,11},</code>
150	<code>{71.54,31334,11}};</code>
151	<code>float hasilhitung[totalDataLatih][2],</code>
152	<code>hasilsort[totalDataLatih][2];</code>
153	<code>void knn (float du[2]) {</code>
154	<code>    //Serial.println(du[1]);</code>
155	<code>    for (size_t i = 0; i &lt; totalDataLatih; i++){</code>
156	<code>        hasilhitung[i][0] = sqrt(abs(pow(datalatih[i][0], 2)-</code>
157	<code>pow(du[0], 2))+abs(pow(datalatih[i][1], 2)-pow(du[1], 2)));</code>
158	<code>        hasilhitung[i][1] = datalatih[i][2];</code>
159	<code>        //Serial.println(hasilhitung[i][0]);</code>
160	<code>        //Serial.println(hasilhitung[i][1]);</code>
161	<code>    }</code>
162	<code>    memcpy(hasilsort, hasilhitung, sizeof(hasilsort));</code>
163	<code>    for (size_t i = 1; i &lt; totalDataLatih; i++) {</code>
164	<code>        for (size_t j = i; j &gt; 0 &amp;&amp; hasilsort[j-1][0] &gt;</code>
165	<code>hasilsort[j][0]; j--) {</code>
166	<code>            float tmp[2];</code>
167	<code>            memcpy(tmp, hasilsort[j-1], sizeof(tmp));</code>
168	<code>            memcpy(hasilsort[j-1], hasilsort[j],</code>
169	<code>sizeof(hasilsort[j-1]));</code>
170	<code>            memcpy(hasilsort[j], tmp, sizeof(hasilsort[j]));</code>
171	<code>        }</code>
172	<code>    }</code>
173	<code>}</code>
174	<code>}</code>
175	<code></code>
176	<code>void setup() {</code>
177	<code>    Serial.begin(9600);</code>
178	<code>    Serial.println("====KNN====");</code>
179	<code></code>
180	<code>}</code>
181	<code></code>
182	<code>void loop() {</code>
183	<code>    float datauji[2] = {37.8,4999};</code>
184	<code>    //Serial.println(totalDataLatih);</code>
185	<code>    knn(datauji);</code>
186	<code></code>
187	<code></code>
188	<code>    for (size_t i = 0; i &lt; totalDataLatih; i++){</code>
189	<code>        Serial.print(hasilhitung[i][0]);</code>
190	<code>        Serial.print(" -&gt; ");</code>
191	<code></code>
192	<code></code>

No	Code Program
193	<code>Serial.print(int(hasilhitung[i][1]));</code>
194	<code>Serial.println(kelas[int(hasilhitung[i][1]))];delay(100);</code>
195	<code>}</code>
196	<code>Serial.println();</code>
197	<code>for (size_t i = 0; i &lt; totalDataLatih; i++){</code>
198	<code>Serial.print("K");</code>
199	<code>Serial.print(i+1);Serial.print(" = ");</code>
200	<code>Serial.print(hasilsort[i][0]);</code>
201	<code>Serial.print(" -&gt; ");</code>
202	<code>Serial.print(int(hasilsort[i][1]));</code>
203	<code>Serial.println(kelas[int(hasilsort[i][1]))];</code>
204	<code>//Serial.println("K"+String(i+1)+" =</code>
205	<code>"+String(hasilsort[i][0])+" -&gt;</code>
206	<code>"+kelas[int(hasilsort[i][1]))];</code>
207	<code>}</code>
208	<code>Serial.println();</code>
209	<code>Serial.print(hasilsort[0][0]);</code>
210	<code>Serial.print(" -&gt; ");</code>
211	<code>Serial.println(kelas[int(hasilsort[0][1]))];</code>
212	<code></code>
213	<code></code>
214	<code>//Serial.println(String(hasilsort[0][0])+" -&gt;</code>
215	<code>"+kelas[int(hasilsort[0][1]))];</code>
216	<code>Serial.println();</code>
217	<code>delay (10000);</code>
218	<code>}</code>
219	<code></code>

## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis membahas tentang proses pengujian pada penelitian tentang “*Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Metode K-Nearest Neighbor (K-NN)*”. Terdapat 4 pengujian yang akan dilakukan, yaitu pengujian akuisisi data sensor getar, akuisisi data *load cell*, akuisisi data *limit switch*, dan proses klasifikasi K-NN. Pengujian tersebut ditunjukkan pada Gambar 6.1.



Gambar 6.1 Pengujian dan Analisis

### 6.1 Pengujian Akuisisi Data Sensor Getar

#### 6.1.1 Tujuan

Tujuan dilakukannya pengujian akuisisi data pada sensor getar adalah untuk mengetahui getaran dari *silhouette* ketika *silhouette* tertembak maupun tersentuh. Pengujian dilakukan dengan 3 mode kepada *silhouette* yakni mode normal(diam), mode tersentuh-sentuh, dan mode tertembak.

#### 6.1.2 Langkah-Langkah

Langkah-Langkah yang harus dilakukan untuk melakukan pengujian pada sensor getar antara lain :

- Merancang Sensor Getar dengan Arduino Mega 2560 menggunakan kabel jumper, sehingga sensor dengan kontroler terhubung melalui pinnya.
- Buka Arduino IDE dan memprogramnya untuk membaca sensor getar
- Compile dan upload kode program yang telah dibuat.

- d. Memberikan masukan pada sensor getar berupa 3 perlakuan yaitu, normal (disentuh tetapi tidak jatuh), disentuh sehingga *silhouette* jatuh, dan ditembak.
- e. Amati hasil output pada sensor getar yang ditampilkan pada serial monitor Arduino IDE, dan mengambil 120 sampel dari hasil pembacaan tersebut.

### 6.1.3 Hasil dan Analisis

Berikut adalah Tabel yang merupakan hasil dari pembacaan sensor getar setelah melalui berbagai macam hasil kelas dari tiap *silhouette* yang didapatkan data dari Sensor Getar.

**Tabel 6.1** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Ayam dengan Kondisi Normal.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Ayam	Normal	932NM
2	Ayam	Normal	4999 NM
3	Ayam	Normal	1840 NM
4	Ayam	Normal	5043 NM
5	Ayam	Normal	6183 NM
6	Ayam	Normal	9507 NM
7	Ayam	Normal	4424 NM
8	Ayam	Normal	4077 NM
9	Ayam	Normal	7647 NM
10	Ayam	Normal	8341 NM
Rata-Rata			5299.3 NM

Dari Data Pengujian pada Tabel 6.1 Menunjukkan bahwa Jenis *silhouette* ayam dengan kondisi normal mempunyai rata-rata getaran sebesar 5299.3 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada tabel diatas.

**Tabel 6.2** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Ayam dengan Kondisi Tersentuh.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Ayam	Tersentuh	21268 NM
2	Ayam	Tersentuh	19670 NM
3	Ayam	Tersentuh	19118 NM
4	Ayam	Tersentuh	17702 NM
5	Ayam	Tersentuh	12560 NM
6	Ayam	Tersentuh	15600 NM
7	Ayam	Tersentuh	22160 NM
8	Ayam	Tersentuh	23938 NM
9	Ayam	Tersentuh	15879 NM
10	Ayam	Tersentuh	17761 NM
Rata-Rata			18565.6 NM

Dari Data Pengujian pada Tabel 6.2 Menunjukkan bahwa Jenis *silhouette* ayam dengan kondisi Tersentuh mempunyai rata-rata getaran sebesar 18565.6 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel diatas.

**Tabel 6.3** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Ayam dengan Kondisi Tertembak.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Ayam	Tertembak	31775 NM
2	Ayam	Tertembak	31255 NM
3	Ayam	Tertembak	25647 NM
4	Ayam	Tertembak	21311 NM
5	Ayam	Tertembak	26102 NM
6	Ayam	Tertembak	26207 NM
7	Ayam	Tertembak	28161 NM
8	Ayam	Tertembak	25008 NM
9	Ayam	Tertembak	21080 NM
10	Ayam	Tertembak	28782 NM
Rata-Rata			26532.8 NM

Dari Data Pengujian pada Tabel 6.3 Menunjukkan bahwa Jenis *silhouette* ayam dengan kondisi Tertembak mempunyai rata-rata getaran sebesar 26532.8 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

**Tabel 6.4** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Kalkun dengan Kondisi Normal.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Kalkun	Normal	8313 NM
2	Kalkun	Normal	12810 NM
3	Kalkun	Normal	6676 NM
4	Kalkun	Normal	5432 NM
5	Kalkun	Normal	7091 NM
6	Kalkun	Normal	6617 NM
7	Kalkun	Normal	3192 NM
8	Kalkun	Normal	6066 NM
9	Kalkun	Normal	8992 NM
10	Kalkun	Normal	7767 NM
Rata-Rata			7295.6 NM

Dari Data Pengujian pada Tabel 6.4 Menunjukkan bahwa Jenis *silhouette* Kalkun dengan kondisi Normal mempunyai rata-rata getaran sebesar 7295.6 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

**Tabel 6.5** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Kalkun dengan Kondisi Tersentuh.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Kalkun	Tersentuh	7240 NM
2	Kalkun	Tersentuh	20640 NM
3	Kalkun	Tersentuh	21292 NM
4	Kalkun	Tersentuh	15156 NM
5	Kalkun	Tersentuh	25484 NM
6	Kalkun	Tersentuh	16805 NM
7	Kalkun	Tersentuh	25082 NM
8	Kalkun	Tersentuh	11193 NM
9	Kalkun	Tersentuh	16202 NM
10	Kalkun	Tersentuh	15748 NM
Rata-Rata			17484.2 NM

Dari Data Pengujian pada Tabel 6.5 Menunjukkan bahwa Jenis *silhouette* Kalkun dengan kondisi Tersentuh mempunyai rata-rata getaran sebesar 17484.2 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel diatas.



**Tabel 6.6** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Kalkun dengan Kondisi Tertembak.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Kalkun	Tertembak	20595 NM
2	Kalkun	Tertembak	30771 NM
3	Kalkun	Tertembak	23048 NM
4	Kalkun	Tertembak	31629 NM
5	Kalkun	Tertembak	24468 NM
6	Kalkun	Tertembak	24770 NM
7	Kalkun	Tertembak	23100 NM
8	Kalkun	Tertembak	24705 NM
9	Kalkun	Tertembak	25169 NM
10	Kalkun	Tertembak	27428 NM
Rata-Rata			25568.3 NM

Dari Data Pengujian pada Tabel 6.6 Menunjukkan bahwa Jenis *silhouette* Kalkun dengan kondisi Tertembak mempunyai rata-rata getaran sebesar 25568.3 Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

**Tabel 6.7** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Babi dengan Kondisi Normal.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Babi	Normal	5101 NM
2	Babi	Normal	6498 NM
3	Babi	Normal	5226 NM
4	Babi	Normal	5969 NM
5	Babi	Normal	5467 NM
6	Babi	Normal	7633 NM
7	Babi	Normal	8930 NM
8	Babi	Normal	8046 NM
9	Babi	Normal	5978 NM
10	Babi	Normal	8431 NM
Rata-Rata			6727.9 NM

Dari Data Pengujian pada Tabel 6.7 Menunjukkan bahwa Jenis *silhouette* Babi dengan kondisi Normal mempunyai rata-rata getaran sebesar 6727.9 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

**Tabel 6.8** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Babi dengan Kondisi Tersentuh.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Babi	Tersentuh	19143 NM
2	Babi	Tersentuh	17091 NM
3	Babi	Tersentuh	21950 NM
4	Babi	Tersentuh	26320 NM
5	Babi	Tersentuh	8295 NM
6	Babi	Tersentuh	31020 NM
7	Babi	Tersentuh	18752 NM
8	Babi	Tersentuh	8107 NM
9	Babi	Tersentuh	11712 NM
10	Babi	Tersentuh	17367 NM
Rata-Rata			17975.7 NM

Dari Data Pengujian pada Tabel 6.8 Menunjukkan bahwa Jenis *silhouette* Babi dengan kondisi Tersentuh mempunyai rata-rata getaran sebesar 17975.7 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

**Tabel 6.9** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Babi dengan Kondisi Tertembak.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Babi	Tertembak	22854 NM
2	Babi	Tertembak	29172 NM
3	Babi	Tertembak	28158 NM
4	Babi	Tertembak	30135 NM
5	Babi	Tertembak	28690 NM
6	Babi	Tertembak	22291 NM
7	Babi	Tertembak	24367 NM
8	Babi	Tertembak	24035 NM
9	Babi	Tertembak	29278 NM
10	Babi	Tertembak	28390 NM
Rata-Rata			26737 NM

Dari Data Pengujian pada Tabel 6.9 Menunjukkan bahwa Jenis *silhouette* Babi dengan kondisi Tertembak mempunyai rata-rata getaran sebesar 26737 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

**Tabel 6.10** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Domba dengan Kondisi Normal.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Domba	Normal	3268 NM
2	Domba	Normal	3100 NM
3	Domba	Normal	3950 NM
4	Domba	Normal	4730 NM
5	Domba	Normal	1973 NM
6	Domba	Normal	4193 NM
7	Domba	Normal	3168 NM
8	Domba	Normal	2228 NM
9	Domba	Normal	5295 NM
10	Domba	Normal	2334 NM
Rata-Rata			3423.9 NM

Dari Data Pengujian pada Tabel 6.10 Menunjukkan bahwa Jenis *silhouette* Domba dengan kondisi Normal mempunyai rata-rata getaran sebesar 3423.9 . Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

**Tabel 6.11** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Domba dengan Kondisi Tersentuh.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Domba	Tersentuh	28717 NM
2	Domba	Tersentuh	6174 NM
3	Domba	Tersentuh	7210 NM
4	Domba	Tersentuh	21913 NM
5	Domba	Tersentuh	9906 NM
6	Domba	Tersentuh	21913 NM
7	Domba	Tersentuh	10070 NM
8	Domba	Tersentuh	16456 NM
9	Domba	Tersentuh	7528 NM
10	Domba	Tersentuh	16251 NM
Rata-Rata			14613.8 NM

Dari Data Pengujian pada Tabel 6.11 Menunjukkan bahwa Jenis *silhouette* Domba dengan kondisi Tersentuh mempunyai rata-rata getaran sebesar 14613.8 Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

**Tabel 6.12** Hasil Pengujian Sensor Getar Dengan Jenis *Silhouette* Domba dengan Kondisi Tertembak.

Percobaan Ke-	Jenis <i>Silhouette</i>	Kondisi	Getaran
1	Domba	Tertembak	24385 NM
2	Domba	Tertembak	26282 NM
3	Domba	Tertembak	31002 NM
4	Domba	Tertembak	30418 NM
5	Domba	Tertembak	29028 NM
6	Domba	Tertembak	29526 NM
7	Domba	Tertembak	27052 NM
8	Domba	Tertembak	24697 NM
9	Domba	Tertembak	25848 NM
10	Domba	Tertembak	31334 NM
Rata-Rata			27957.2 NM

Dari Data Pengujian pada Tabel 6.12 Menunjukkan bahwa Jenis *silhouette* Domba dengan kondisi Tertembak mempunyai rata-rata getaran sebesar 27957.2 Yang dimana hal tersebut menjadi acuan output sistem apabila dengan kondisi target dan getaran mendekati hasil pada Tabel di atas.

Hasil Analisa dari tiap-tiap *silhouette* yakni Ayam , Kalkun , Babi dan Domba dengan masing masing kondisi normal , tersentuh dan tertembak pada tabel di atas didapatkan bahwa ketika *silhouette* dengan kondisi normal nilai getaran yang di hasilkan cenderung rendah yakni sekitar (900-12900) dikarenakan kondisi *silhouette* diam tidak terkena tembakan peluru. Ketika *silhouette* dengan kondisi tersentuh maka nilai getaran yang di hasilkan akan sedikit lebih tinggi yakni sekitar (13000-19900) dikarenakan tembakan mengenai target *silhouette* namun tidak sampai terjatuh. Sedangkan saat *silhouette* dengan kondisi tertembak maka getaran yang di hasilkan cenderung tinggi yakni (>20000) yang disebabkan target *silhouette* tertembak dan target terjatuh.

## 6.2 Pengujian Akuisisi Data *Load Cell*

### 6.2.1 Tujuan

Tujuan dilakukannya pengujian akuisisi data pada *load cell* adalah untuk mengetahui berat dari masing-masing jenis *silhouette*.

### 6.2.2 Langkah-Langkah

Prosedur yang harus dilakukan untuk melakukan pengujian pada *load cell* antara lain :

- Merancang *load cell* dengan Arduino Mega 2560 menggunakan kabel jumper, sehingga *load cell* dengan kontroler terhubung melalui pinnya.
- Buka Arduino IDE dan memprogramnya untuk membaca nilai berat pada *load cell*.
- Compile dan upload kode program yang telah dibuat.
- Memberikan masukan pada *load cell* berupa jenis *silhouette* yang berbedabeda yaitu *silhouette* jenis ayam, kalkun, babi, dan domba.
- Amati hasil output pada *load cell* yang ditampilkan pada serial monitor Arduino IDE. Kemudian mengambil 40 sampel dari *load cell* dan timbangan
- Menentukan Sistem Error (%) dengan rumus :

$$\text{Sistem Error}(\%) = \frac{[PL - PT]}{PL} \times 100$$

Keterangan :

PL : Pembacaan Nilai Pada *Load Cell*

PT : Pembacaan Nilai Pada Timbangan

- Menentukan rata-rata dari Sistem Error(%) dengan rumus :

$$\text{Rata - rata} = \frac{\text{Jumlah Hasil}}{\text{Banyaknya Percobaan}}$$

### 6.2.3 Hasil dan Analisis

Berikut adalah Tabel yang merupakan hasil dari pembacaan nilai berat pada *load cell*.

**Tabel 6.13** Hasil Pengujian *Load Cell* pada *Silhouette* Ayam

Percobaan Ke-	Jenis <i>Silhouette</i>	Pembacaan <i>Load Cell</i> (Gram)	Pembacaan Timbangan (Gram)	Error %
1	Ayam	31.94	29	9.2
2	Ayam	30.6	29	5.23
3	Ayam	33.62	29	13.74
4	Ayam	30.16	29	3.85
5	Ayam	31.9	29	9.09
6	Ayam	27.63	29	4.96
7	Ayam	32.12	29	9.71
8	Ayam	27.32	29	6.15
9	Ayam	26.38	29	9.93
10	Ayam	27.38	29	5.92
Rata-Rata		29.905	29	7.778

Dari Data Pengujian pada Tabel 6.13 Menunjukkan bahwa Jenis silhouette Ayam mempunyai berat pada timbangan sebesar 29 gram. Dan Berat rata-rata dalam pembacaan *load cell* sebesar 29.905 gram dengan *error* berat rata-rata *load cell* sebesar 7.778%.

**Tabel 6.14** Hasil Pengujian *Load Cell* pada *Silhouette* Kalkun

Percobaan Ke-	Jenis <i>Silhouette</i>	Pembacaan <i>Load Cell</i> (Gram)	Pembacaan Timbangan (Gram)	Error %
1	Kalkun	35.8	36	0.56
2	Kalkun	36.66	36	1.8
3	Kalkun	35.36	36	1.81
4	Kalkun	35.74	36	0.73
5	Kalkun	35.25	36	2.13
6	Kalkun	36.15	36	0.41
7	Kalkun	36.83	36	2.25
8	Kalkun	33.99	36	5.91
9	Kalkun	36.62	36	1.69
10	Kalkun	36.04	36	0.11
Rata-Rata		35.844	36	1.74

Dari Data Pengujian pada Tabel 6.14 Menunjukkan bahwa Jenis silhouette Kalkun mempunyai berat pada timbangan sebesar 36 gram. Dan Berat rata-rata dalam pembacaan *load cell* sebesar 35.844 gram dengan *error* berat rata-rata *load cell* sebesar 1.74%.

**Tabel 6.15** Hasil Pengujian *Load Cell* pada *Silhouette* Babi

Percobaan Ke-	Jenis <i>Silhouette</i>	Pembacaan <i>Load Cell</i> (Gram)	Pembacaan Timbangan (Gram)	Error %
1	Babi	46.9	48	2.35
2	Babi	47.27	48	1.54
3	Babi	49.38	48	2.79
4	Babi	47.27	48	1.54
5	Babi	47.27	48	1.54
6	Babi	46.4	48	3.45
7	Babi	43.36	48	10.7
8	Babi	46.9	48	2.35
9	Babi	48.29	48	0.6
10	Babi	47	48	2.13
Rata-Rata		47.004	48	2.899



Dari Data Pengujian pada Tabel 6.15 Menunjukkan bahwa Jenis *silhouette* Kalkun mempunyai berat pada timbangan sebesar 48 gram. Dan Berat rata-rata dalam pembacaan *load cell* sebesar 47.004 gram dengan *error* berat rata-rata *load cell* sebesar 2.899%.

**Tabel 6.16** Hasil Pengujian *Load Cell* pada *Silhouette* Babi

Percobaan Ke-	Jenis <i>Silhouette</i>	Pembacaan <i>Load Cell</i> (Gram)	Pembacaan Timbangan (Gram)	Error %
1	Domba	59.95	68	13.43
2	Domba	68.49	68	0.72
3	Domba	62.65	68	8.54
4	Domba	53.2	68	27.82
5	Domba	63.21	68	7.58
6	Domba	68.12	68	0.18
7	Domba	64	68	6.25
8	Domba	70.09	68	2.98
9	Domba	74.2	68	8.36
10	Domba	67.22	68	1.16
Rata-Rata		65.113	68	7.702

Dari Data Pengujian pada Tabel 6.16 Menunjukkan bahwa Jenis *silhouette* Kalkun mempunyai berat pada timbangan sebesar 68 gram. Dan Berat rata-rata dalam pembacaan *load cell* sebesar 65.113 gram dengan *error* berat rata-rata *load cell* sebesar 7.702%.

Hasil Analisa dari tiap-tiap *silhouette* yakni Ayam , Kalkun , Babi dan Domba terdapat nilai eror dalam pengujiannya bahwa hasil berat tiap-tiap *silhouette* telat melewati beberapa prosedur atau langkah-langkah sehingga didapatkan data dari *load cell* dan timbangan digital sebagai parameter pembandingan.

### 6.3 Pengujian Akuisisi Data *Limit Switch*

#### 6.3.1 Tujuan Pengujian

Tujuan dilakukannya pengujian akuisisi data pada *limit switch* adalah untuk mengetahui apakah *silhouette* jatuh atau tidak.

### 6.3.2 Langkah-Langkah

Langkah-Langkah yang harus dilakukan untuk melakukan pengujian pada *limit switch* diantaranya sebagai berikut :

1. Merancang *limit switch* dengan rangkaian debounce dan dihubungkan ke Arduino Mega 2560 menggunakan kabel jumper yang disambungkan ke pinnya.
2. Buka Arduino IDE dan memprogramnya untuk membaca nilai pada *limit switch*.
3. Compile dan upload kode program yang telah dibuat.
4. Memberikan masukan pada *limit switch* berupa *silhouette* terjatuh atau tidak.
5. Amati hasil output pada *limit switch* yang ditampilkan pada serial monitor Arduino IDE.
6. Kesimpulan.

### 6.3.3 Hasil dan Analisis

Berikut adalah Tabel yang merupakan hasil dari pembacaan value pada *limit switch*.

**Tabel 6.17** Hasil Pengujian *Limit Switch*

Percobaan Ke-	Kondisi	Pembacaan <i>Limit Switch</i>
1	<i>Silhouette</i> Terjatuh	1
2	<i>Silhouette</i> Terjatuh	1
3	<i>Silhouette</i> Terjatuh	1
4	<i>Silhouette</i> Normal	0
5	<i>Silhouette</i> Normal	0
6	<i>Silhouette</i> Normal	0

Pada Tabel 6.17 yang merupakan hasil dari pembacaan dari *limit switch* setelah melalui berbagai macam prosedur sehingga berhasil didapatkan data dari *Limit Switch* . Hal ini menunjukkan analisa bahwa ketika *Silhouette* terjatuh, maka data yang di baca adalah 1, dan ketika *Silhouette* tidak jatuh maka data yang dibaca adalah 0.

## 6.4 Pengujian Proses Klasifikasi *K-Nearest Neighbor*

### 6.4.1 Tujuan Pengujian

Tujuan dari dilakukannya pengujian pada proses klasifikasi *K-Nearest Neighbor* untuk mengetahui keberhasilan sistem agar dapat menentukan skor pada perlombaan menembak.

### 6.4.2 Langkah-Langkah

Langkah-Langkah yang harus dilakukan untuk melakukan pengujian Klasifikasi *K-Nearest Neighbor* antara lain :

1. Merancang *load cell*, sensor getar, *limit switch*, Arduino Mega 2560(1), dan Arduino Mega 2560(2) dengan menggunakan kabel jumper.
2. Buka Arduino IDE dan memprogram klasifikasi *K-Nearest Neighbor* pada pusat kontroler (Arduino Mega 2560(1) dan memprogram inisialisasi tiap target *silhouette* pada control input (Arduino Mega 2560 (2).
3. Compile dan upload kode program yang telah dibuat.
4. Buka serial monitor pada Arduino IDE yang bertujuan untuk mengamati output dari inisialisasi tiap target.
5. Memberikan input pada masing-masing sensor yang terpasang pada *silhouette*, sehingga sensor dapat mengirimkan masukan ke Kontroler input (Arduino Mega 2560) untuk dikirimkan lagi ke pusat kontroler (Arduino Mega) dan memproses data input tersebut menggunakan Klasifikasi *K-Nearest Neighbor*.
6. Buka Aplikasi Delphi untuk di connectkan ke Arduino IDE Kontroler pusat sebagai tampilan output untuk mengetahui hasil target *silhouette* yang di peroleh dari Klasifikasi *K-Nearest Neighbor*.
7. Amati hasil yang ditampilkan pada Aplikasi Delphi dan mengambil 12 sampel. Output yang dihasilkan pada sistem mempresentasikan beberapa kondisi yaitu ayam normal, ayam tersentuh, ayam tertembak, kalkun normal, kalkun tersentuh, kalkun tertembak, babi normal, babi tersentuh, babi tertembak, domba normal, domba tersentuh dan domba tertembak.
8. Melakukan perhitungan klasifikasi *K-Nearest Neighbor* dari pembacaan sensor getar dan *load cell* secara manual dan membandingkannya dengan hasil didapatkan pada Aplikasi Delphi.

### 6.4.3 Hasil dan Analisis

Pada Tabel 6.18 dibawah ini dengan menggunakan 12 sampel dengan input dan output yang berbeda-beda yaitu ayam tersentuh, ayam tertembak, kalkun tersentuh, kalkun tertembak, babi tersentuh, babi tertembak, domba tersentuh, dan domba tertembak. Pengujian dilakukan dengan menganalisis dari rumus klasifikasi *K-Nearest Neighbor* yang telah diterapkan pada sistem yang akan memberikan keputusan output sesuai dengan perhitungan Klasifikasi *K-Nearest*

*Neighbor*. Dari 12 sampel dengan input yang berbeda-beda akan memberikan output sistem sesuai dengan perancangan. Hal tersebut dibuktikan dengan perhitungan manual yang menunjukkan hasil yang sama. Berikut merupakan pengujian menggunakan perhitungan manual yang berdasarkan perhitungan klasifikasi *K-Nearest Neighbor*. Berikut adalah data acak yang akan di ujikan secara manual dan system :

- $y_1$  Berat (Data yang Uji) : 42
- $y_2$  Getar (Data yang Uji) : 23074
- $x_1$  Berat (Data Training) : Terdapat pada Tabel 6.13-6.16
- $x_2$  Getar (Data Training) : Terdapat pada Tabel 6.1-6.12

Menentukan Parameter K ( Jumlah tetangga paling dekat ) Dengan Melakukan testing pada data training dan nilai K dari data yang coba diujikan. Dengan Rumus sbg berikut :

$$[\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}] =$$

- Menghitung Kuadrat jarak Euclid . Setelah data di masukkan ke masing – masing variable yakni parameter  $K = 1 - K = 120$  (Total dari Data Training)
- Mengurutkan Objek-objek yang memiliki nilai terkecil hingga nilai terbesar.
- Hasil Output berupa Nilai Kelas dari klasifikasi *K-Nearest Neighbor*.

```

K100 = 22340.83 -> 6Babi Normal
K101 = 22349.36 -> 3Kalkun Normal
K102 = 22382.26 -> 9Domba Normal
K103 = 22398.42 -> 6Babi Normal
K104 = 22427.22 -> 6Babi Normal
K105 = 22440.34 -> 0Ayam Normal
K106 = 22450.17 -> 0Ayam Normal
K107 = 22508.45 -> 9Domba Normal
K108 = 22570.54 -> 0Ayam Normal
K109 = 22614.63 -> 9Domba Normal
K110 = 22635.79 -> 0Ayam Normal
K111 = 22658.32 -> 9Domba Normal
K112 = 22766.73 -> 9Domba Normal
K113 = 22777.44 -> 3Kalkun Normal
K114 = 22780.83 -> 9Domba Normal
K115 = 22790.22 -> 9Domba Normal
K116 = 22881.34 -> 9Domba Normal
K117 = 22891.89 -> 9Domba Normal
K118 = 22915.28 -> 9Domba Normal
K119 = 22926.30 -> 0Ayam Normal
K120 = 22981.13 -> 0Ayam Normal

1486.85 -> Kalkun Tertembak

```

**Gambar 6.2** Hasil Pengujian *K-Nearest Neighbor* Dari Data Uji Diatas

Pada Gambar 6.2 menunjukkan bahwa output pada sistem adalah Kalkun Tertembak . Ouput yang dihasilkan pada perhitungan manual sesuai dengan perhitungan sistem. Hal tersebut menunjukkan bahwa sistem sudah sesuai dengan mengeluarkan output sistem yang sesuai pada perancangan dan sudah memenuhi kebutuhan fungsional sistem. Berikut merupakan hasil analisis pengujian tingkat keakuratan klasifikasi *K-Nearest Neighbor* yang ditunjukkan pada Tabel 6.18 di bawah ini.

**Tabel 6.18** Hasil Pengujian *K-Nearest Neighbor*

Percobaan Ke-	Pembacaan Load Cell	Pembacaan Sensor Getar	Kondisi Silhouette	Output Secara Manual	Output Secara Sistem	Persentase Manual dengan Sistem
1	38.21	26687	Kalkun Tertembak	SESUAI (1)	SESUAI (1)	100%
2	58.31	30480	Domba Tertembak	SESUAI (1)	SESUAI (1)	100%
3	59.09	18124	Domba Tersentuh	SESUAI (1)	SESUAI (1)	100%
4	34.61	13901	Kalkun Tersentuh	SESUAI (1)	SESUAI (1)	100%
5	48.02	14092	Babi Tersentuh	SESUAI (1)	SESUAI (1)	100%
6	31.9	6802	Ayam Normal	SESUAI (1)	SESUAI (1)	100%
7	33.9	31474	Ayam Tertembak	SESUAI (1)	SESUAI (1)	100%
8	37.22	5403	Kalkun Normal	SESUAI (1)	SESUAI (1)	100%
9	47.80	6230	Babi Normal	SESUAI (1)	SESUAI (1)	100%
10	32.7	14072	Ayam Tersentuh	SESUAI (1)	SESUAI (1)	100%
11	48.21	32654	Babi Tertembak	SESUAI (1)	SESUAI (1)	100%
12	59.30	64213	Domba Normal	SESUAI (1)	SESUAI (1)	100%

Pada Tabel 6.18 Diatas menunjukkan bahwa Pengujian *K-Nearest Neighbor* (*K-NN*) dengan berat dan getaran yang di ujikan sudah sesuai , dibuktikan dengan output secara sistem yang bernilai 1 dan perhitungan manual bernilai 1 yang di artikan sesuai dengan output yang di diharapkan. Maka dengan demikian Pengujian *K-Nearest Neighbor* (*K-NN*) berhasil dengan keakuratan sebesar 100%.



## BAB 7 PENUTUP

Pada bab ini membahas kesimpulan dari penelitian “*Scoring System Otomatis Pada Perlombaan Menembak Dengan Target Silhouette Hewan Menggunakan Metode Klasifikasi K-Nearest Neighbor*” dan saran untuk pengembangan topik skripsi atau penelitian lebih lanjut.

### 7.1 Kesimpulan

Kesimpulan yang didapatkan berdasarkan hasil dari pengujian dan analisa yang telah dilakukan terhadap penelitian ini adalah sebagai berikut.

1. Proses perancangan sistem scoring pada perlombaan menembak dengan target yang berupa *silhouette* hewan dengan menyambungkan *load cell*, sensor getar, dan *limit switch* ke kontroler input (Arduino Mega 2560(1)) dan menghubungkan kontroler input (Arduino Mega 2560(1)) ke pusat kontroler (Arduino Mega 2560(2)) menjadi sebuah sistem yang saling terhubung satu sama lain. Proses akuisisi pada *load cell* meliputi pembacaan berat pada *silhouette*. Akuisisi pada sensor getar meliputi pembacaan getaran pada *silhouette*. Sedangkan proses akuisisi data pada *limit switch* meliputi pembacaan value ketika *silhouette* terjatuh atau tidak.
2. Pada penelitian ini telah dibuat “*Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Metode K-Nearest Neighbor*”. Semua komponen alat yang digunakan maupun metode *Klasifikasi K-Nearest Neighbor* itu sendiri dapat diterapkan digunakan sesuai dengan yang diinginkan pengguna, terbukti dengan sistem dapat mengklasifikasikan jenis *Silhouette* yang diam/normal, tersentuh maupun tertembak.
3. Pengujian *K-Nearest Neighbor* dilakukan dengan melakukan perbandingan antara perhitungan pada sistem dan perhitungan manual. Pada hasil perhitungan tersebut menunjukkan hasil yang sama. Hal tersebut menunjukkan bahwa hasil dari pengujian menggunakan Metode Klasifikasi *K-Nearest Neighbor* pada sistem sesuai dengan perancangan yang dibuat. Sistem scoring ini dapat menentukan berbagai kondisi output yaitu ayam normal, ayam tersentuh, ayam tertembak, kalkun normal, kalkun tersentuh, kalkun tertembak, babi normal, babi tersentuh, babi tertembak, domba normal, domba tersentuh dan domba tertembak dengan input berupa berat dan getaran yang bervariasi. Pada pengujian *K-Nearest Neighbor* yang didapatkan persentase dari keberhasilan sistem sebesar 100%. Hal ini menunjukkan bahwa sistem scoring dapat bekerja dengan baik dalam menentukan kondisi *Silhouette* yang akan mempengaruhi skor pada perlombaan menembak.



## 7.2 Saran

Berdasarkan pembuatan dan hasil pada pengujian “*Scoring System* Otomatis Pada Lomba Menembak Dengan Target *Sillhouette* Hewan Menggunakan Metode *K-Nearest Neighbor* (K-NN)”, dapat diberikan saran untuk pengembangan penelitian selanjutnya, antara lain:

1. Menggunakan *load cell* dan sensor getar yang memiliki sensitivitas yang tinggi supaya hasil yang di dapatkan lebih akurat lagi.
2. Sistem dapat mengirimkan data antara kontroler input dan pusat kontroler menggunakan wireless (memakai NRF).
3. Output sistem dapat ditampilkan melalui website (IoT).
4. Dilakukan Perbandingan dengan penelitian sekarang yang sudah dilakukan dengan sebelumnya yakni menggunakan Logika Fuzzy , Metode *Naïve Bayes* dan Metode *K-Nearest Neighbor*. Dari ketiga metode tersebut manakah yang lebih akurat dalam pengambil data maupun output yang di hasilkan serta apa kekurangan dan kelebihan dari tiap – tiap klasifikasi yang digunakan.



## DAFTAR PUSTAKA

- Bronshtein, A. (2017, April 11). *A Quick Introduction to K-Nearest Neighbors Algorithm*. Diambil kembali dari Medium: <https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7>
- Depok Instruments. (2011, July 20). *ADC (Analog to Digital Converter)*. Diambil kembali dari Depok Instruments: <https://depokinstruments.com/2011/07/20/adc-analog-to-digital-converter/>
- Ecadio. (t.thn.). *Mengenal Arduino Mega 2560*. Diambil kembali dari Ecadio: <http://ecadio.com/belajar-dan-mengenal-arduino-mega>
- Galco. (2018). *Limit Switch*. Diambil kembali dari Galco: <http://www.galco.com/comp/prod/limitswi.htm>
- Harmaen, H. (2012). Rancang Bangun Sisem Penghitung Skor Otomatis Olahraga Menembak Menggunakan Algoritma Tempalte Matching. *Design and Implementation Automatic Scoring Computation System for Shooting Sport with Template Matching Algorithm*, 6.
- Pelana, R. &. (2013). Sport Science. *Alat Penampil Skor Otomatis Pada Papan Target Dalam Olahraga Panahan*, 39-46.
- Ramadhan, H. (2018). Scoring System Otomatis pada Lomba Menembak dengan Target Silhouette Hewan Menggunakan Logika Fuzzy. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2625-2634.
- S. Ying, W. G. (2011). The Application of Information Technology in Sports Training. *International Conference on Future Computer Science and Education*, 210-212.
- Sport Tourism. (2018, January 15). *Olahraga Menembak Bisa Melatih Konsentrasi dan Kesabaran*. Diambil kembali dari Sport Tourism: <https://sporttourism.id/sport/olahraga-menembak-bisa-melatih-konsentrasi-dan-kesabaran>
- Theory Circuit. (2017, September 18). *SW-420 Vibration Sensor Arduino Interface*. Diambil kembali dari Theory Circuit: <http://www.theorycircuit.com/sw-420-vibration-sensor-arduino-interface/>
- Tjahjadi, C. (2018, June 09). *Saklar Bounce dan Debounce*. Diambil kembali dari Christianto Tjahjadi: <http://christianto.tjahyadi.com/mikrokontroler/bounce-dan-debounce.html>
- Wardhani, K. (2005). STIKOM. *Sistem Pendukung Keputusan Kelayakan Kredit Pemilikan Mobil dengan Metode Scoring System*.